PDP-8 COMPUTER INTERFACES FOR TROPOSPHERIC PROPAGATION MEASUREMENTS

James E. Crapuchettes

Stanford University

Prepared for:

Joint Services Electronics Program

March 1973

SEL-73-007

PDP-8 Computer Interfaces for Tropospheric Propagation Measurements

by

James E. Crapuchettes

March 1973

Technical Report No. 4506-1

This work was supported in part by the
Joint Services Electronics Program
(U.S. Army, U.S. Navy, and U.S. Air Force)
Contract N00014-67-A-0112-0044

RADIOSCIENCE LABORATORY

STANFORD ELECTRONICS LABORATORIES

STANFORD UNIVERSITY · STANFORD, CALIFORNIA

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author) Stanford Electronics Laboratories Stanford University Stanford, California 94305	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3 REPORT TITLE

PDP-8 COMPUTER INTERFACES FOR TROPOSPHERIC PROPAGATION MEASUREMENTS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical Report No. 4506-1, March 1973

5 AUTHOR(S) (First name, middle initial, last name)

James E. Crapuchettes

6 REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
March 1973	79	8

8a. CONTRACT OR GRANT NO	9a. ORIGINATOR'S REPORT NUMBER(S)
N00014-67-A-0112-0044 b. PROJECT NO	TR No. 4506-1 SEL-73-007
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10 DISTRIBUTION STATEMENT

Reproduction in whole or in part is permitted for any purpose of the United States Government. This document has been approved for public release and sale; its distribution is unlimited.

11 SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	U.S. Army, U.S. Navy, U.S. Air Force

13 ABSTRACT

Interfaces for a PDP-8 computer used in Tropospheric propagation research have been developed to provide communication with and control over the real-time radio equipment. This report describes the work that was done on these interfaces and provides an understanding of their operation.

The characteristics of the Input/Output bus of the PDP-8 computer and the digital circuit modules used in the implementation of the interfaces are provided as background for the discussions of the interfaces.

Each of the interfaces is described with at least three sections: a general description of its function, a specific description of the Input/Output commands used by it and the functions they perform, and a discussion of the interface as it is implemented. In those cases where it is useful, a discussion of the features of or problems with the design and implementation of the interface is also included to aid in the understanding of the interface.

A brief conclusion describes the performance of the finished hardware and the software that was developed for it.

DD FORM 1 NOV 68 1473

UNCLASSIFIED

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computer Interfaces PDP-8 Computer Real-time Control						

ia

PDP-8 COMPUTER INTERFACES FOR TROPOSPHERIC PROPAGATION MEASUREMENTS

by

James E. Crapuchettes

March 1973

Technical Report No. 4506-1

This work was supported in part by the
Joint Services Electronics Program
(U.S. Army, U.S. Navy, and U.S. Air Force)
Contract N00014-67-A-0112-0044

Radioscience Laboratory
Stanford Electronics Laboratories
Stanford University Stanford, California

ABSTRACT

Interfaces for a PDP-8 computer used in Tropospheric propagation research have been developed to provide communication with and control over the real-time radio equipment. This report describes the work that was done on these interfaces and provides an understanding of their operation.

The characteristics of the Input/Output bus of the PDP-8 computer and the digital circuit modules used in the implementation of the interfaces are provided as background for the discussions of the interfaces.

Each of the interfaces is described with at least three sections: a general description of its function, a specific description of the Input/Output commands used by it and the functions they perform, and a discussion of the interface as it is implemented. In those cases where it is useful, a discussion of the features of or problems with the design and implementation of the interface is also included to aid in the understanding of the interface.

A brief conclusion describes the performance of the finished hardware and the software that was developed for it.

TABLE OF CONTENTS

		Page
I.	Introduction	1
II.	Background Information	7
	PDP-8 Programmed Input/Output Bus	7
	Flip-Chip Module Characteristics	15
III.	Hardware Descriptions	24
	Relay Register	27
	Switch Registers	36
	Utility Buffer	42
	120 Hertz Clock	47
	Variable Clock	54
	Event Channels	61
	Plotter Control and Utility Pulses	66
IV.	Conclusions	73
	Appendix A	75
	Appendix B	76
	Appendix C	77
	Appendix D	78
	Bibliography	79

LIST OF ILLUSTRATIONS

Figure		Page
1.	PDP-8 Programmed I/O Bus Block Diagram	9
2.	PDP-8 IOP Generator Logic	12
3.	Multiple-Input Diode-Transistor Gate	17
4.	Diode-Capacitor-Diode Gate	19
5.	Type W103 Device Selector Module	22
6.	System Block Diagram	25
7.	Block Diagram of Stanford Peripherals	26
8.	Relay Register Block Diagram	28
9.	Relay Register Control Section schematic	30
10.	Relay Register Register Section schematic	34
11.	Switch Register Block Diagram	37
12.	Switch Register Control Section schematic	39
13.	Switch Register Switch Box schematic	41
14.	Utility Buffer Block Diagram	43
15.	Utility Buffer schematic	45
16.	120 Hertz Clock Block Diagram	48
17.	120 Hertz Clock Source schematic	50
18.	120 Hertz Clock schematic	52
19.	Variable Clock Block Diagram	55
20.	Variable Clock schematic	58
21.	Event Channels Block Diagram	62
22.	Event Channels schematic	64

ILLUSTRATIONS (continued)

Figure		Page
23.	Plotter Control & Utility Pulses Block Diagram	67
24.	Plotter Control & Utility Pulses schematic	69
25.	X-Y Booster Amplifiers & Plotter Connector schematic	72

APPENDICES

Appendix		Page
A.	Module Utilization Drawing	75
B.	Clock & Event Channel Control Box--part 1	76
C.	Clock & Event Channel Control Box--part 2	77
D.	Clock & Event Channel Control Box--part 3	78

ACKNOWLEDGMENT

I want to especially thank two people for their help with this thesis. My wife, Mary, spent many hours proofreading the drafts--not because she had an interest in or understanding of the subject matter but because she is interested in and understands me. A man can ask for no more.

I also want to thank Agnes Barry for the time that she spent typing and otherwise preparing this thesis for publication. Her good work should not go unnoticed.

I. INTRODUCTION

The Tropospheric Propagation Group at Stanford Electronics Laboratories, under the direction of Dr. Alan Waterman, is engaged in the study of the atmosphere through the use of remote probing by waves (radio or acoustic) propagated through it. This probing is typically done by equipment which consists of one or more transmitters and a number of receivers (referred to as a receiving array). The receivers (the elements of the array) are oriented in a line which is perpendicular to a line from the transmitter(s) to the middle of the receiving array. A wave is launched by a transmitter, passes through the medium (the atmosphere) along the propagation path and is sampled by the receiving array. The amplitude and phase (measured with respect to some signal used as the phase reference) of the output from each receiver in the array is switched by a multiplexer (analog switch) to an Analog-to-Digital (A-D) converter which digitizes the signals. The digitized signals are then read into the computer to which the converter is attached, a PDP-8 manufactured by Digital Equipment Corporation. The signals are then stored in the memory of the computer (queued) and periodically written out onto a magnetic tape for storage. The data stored on the tape is later processed by several different analysis programs to determine how the atmosphere has distorted the wavefront. The purpose is to provide some insight into the processes occurring in the atmosphere at the time that the data was acquired.

There are at present three arrays corresponding to three separate projects (two radio and one acoustic). One of these arrays is used

at a time, controlled by the computer through the multiplexer. This multiplexer switches the array element outputs to produce a switched array output which is the input to the amplitude and phase detectors mentioned above. This multiplexer may be one of two devices, depending on the array in use. One of the devices, used by one of the radio arrays and the acoustic array, may be reset to the first element (element zero) of the array, set to some specific element, or incremented to the next element. The other device, used by the other radio array, may be reset to the first element of the array or incremented to the next element.

In the past, because the characteristics of the array multiplexer depended on the array in use, two separate computer programs were used to control and acquire data from the arrays. These programs were closely tailored to the characteristics of the multiplexer and array(s) with which they were used. The result of this was that when improvements were necessary they were compounded by the need to improve both programs. Added to this was the desirability of acquiring data from the arrays at higher rates than had been used previously and in addition from various meteorological instruments. Consequently, it was decided that a new program was needed which would be versatile enough and fast enough to acquire data from any one of the three arrays and other instruments at the necessary rates. The single program would also help ease the transition to the use of IBM compatible magnetic tape when this equipment became operational.

As the design of the program progressed, it became apparent that the program would use almost all of the input and output devices connected to the computer and that some additional outputs from the program were

also required. These added outputs would be used for transmitter switching, for oscilloscope triggering at the beginning of each scan (sampling of amplitude and phase from each element) of the receiving array and for event information for analog strip-chart recorders. This last output was required to help correlate data recorded on the charts with the data that was actually acquired by the program. Due to this problem, the devices connected to the computer were studied to determine whether any of them could be expanded in some way to provide the outputs that were required. Two of the devices looked as though they could be expanded, so they were studied more closely. These two devices were the control part of the array multiplexer which could be set to any element and the Digital-to-Analog (D-A) converters contained in the D.E.C. Type 34D oscilloscope display.

The array multiplexer control logic, referred to as the Utility Buffer (UB), consisted of a four flip-flop up-counter, the outputs of which were buffered through level converters (saturating amplifiers) and then sent to the actual switches in the arrays. Since the four-bit counter has 16 possible states, the outputs of the counter were used to select one out of 16 or fewer receivers through the switches at the arrays. As mentioned before, this up-counter could be reset to 0, set to some value in the range 0 through 15, or incremented (up-counted) under program control from the computer. Since the length of the word in the PDP-8 is 12 bits, the counter could have been expanded to 12 bits without adding any additional control logic.

The D-A converters each consisted of control logic and a 10 bit register driving an analog ladder switch and operational amplifier.

Used together, they could select one point in a 1024 x 1024 point dot matrix. When used in this manner, a program can display anything on the face of an oscilloscope that can be drawn one point at a time. Again, since the computer word length is 12 bits, the two 10 bit registers could have each been extended by 2 bits without additional control logic.

It soon became clear that attempting to extend the registers in either of these devices might just add more problems. The new outputs would not be independent from the previous outputs of the devices without the addition of a considerable amount of logic. Furthermore, even with modifications, the new outputs would leave much to be desired in versatility and would just barely satisfy the requirements while leaving absolutely no room for future needs. It was therefore decided that a new device should be designed which would be able to provide the additional program outputs in the best way possible and also provide room for future developments. The new device is the 12 bit Relay Register which is described in Chapter 3. The strong points of the device include independent setting or clearing of one or more of the bits in the register and the ability to read the contents of the register back into the computer. Since the bits may also be cleared by push-button switches, the device can also be used for program interaction with users.

During the time in which the new register and control was being designed, the mounting panel in which several other Stanford-built devices were wired was studied to see where the new device could be added. Since these devices had evolved rather unsystematically during the four years that the computer had been in use, the available room was not well used

and it would have been very difficult to install the new device. Therefore a new mounting panel was purchased so that both the old and new devices could be wired into it. This had the added advantage of allowing the wiring and part of the debugging of the new device to be done before the device was installed, minimizing down-time on the heavily used computer. Furthermore, since all of the old devices also had to be wired into the panel, this provided an opportunity for each of the old devices to be studied and redesigned in those cases where the design could be improved.

Finally, some time after the devices were installed and checked out, experience with the data acquisition program showed that some additional switch registers would make the operation of the program (and perhaps others also) more efficient. The implementation chosen, which is also described in Chapter 3, consisted of three 12-bit switch registers which were mounted directly under the console switch register on the computer. They are connected as an input device which can be addressed in a manner similar to the way that the computer console switch register is used.

The following chapter describes the features of the PDP-8 computer which are necessary for an understanding of the devices described in Chapter 3. Included is a discussion of the input/output signal bussing used by the computer and a discussion of some of the logic modules used in the devices described in Chapter 3. The latter discussion pays particular attention to the features of these modules which make their operation different from the operation of today's most commonly used logic.

Chapter 3 describes each of the devices designed and built at Stanford that are added to the input/output bus of the computer. The primary purpose of the chapter is to provide a complete enough description of the devices to be an aid in finding the problem should something go wrong or in modifying the devices, should the need arise in the future. Besides the description of the device as it currently exists, comments have been added describing the devices as originally designed, the problems that resulted from the original design and the steps that were taken to solve these problems.

The final chapter is a brief description of the finished hardware-software system and its operation.

II. BACKGROUND INFORMATION

The equipment descriptions in the next chapter assume a knowledge of the characteristics of the programmed input/output bus of the PDP-8 computer and of the digital modules used in the equipment, Flip-Chip series discrete diode-transistor logic modules. Both the PDP-8 computer and the Flip-Chip logic modules are manufactured by Digital Equipment Corporation. For those not familiar with this computer and these modules, this chapter will provide brief descriptions of the important points of each for a better understanding of the next chapter.

The chapter is divided into two sections. The first section deals with the programmed input/output bus of the PDP-8. Included are descriptions of the physical and electrical characteristics of the bus and of the uses of the signals on the bus. The second section deals with some of the important characteristics of Flip-Chip modules, including the Diode-Capacitor-Diode gate found in some of the modules. Also, a short description of the Type W103 "Device Selector Module," which is used in every device, is included.

1. PDP-8 Programmed Input/Output Bus

The Input/Output (I/O) bus of the PDP-8 computer consists of parallel data and control lines which originate in the PDP-8 processor and go from each peripheral device to the next in a daisy-chain fashion. All peripheral devices are connected in parallel on this bus.

The lines on the bus logically belong to one of two groups, the

programmed I/O bus and the data break (or direct memory access) bus. The data break bus is used for the direct transfer of data between the computer's memory and high data rate peripheral devices. Since this bus is not used by the devices described in the next chapter, it will not be discussed further here. A more complete description of this bus is provided in a Digital Equipment handbook.[1]

The programmed I/O bus (which will be referred to simply as the "I/O bus" from now on), logically consists of a group of input lines and a group of output lines (where "input" and "output" are as seen from the PDP-8 processor). The input lines include three control lines and 12 data lines. The output lines include 22 control lines and 12 data lines. The lines, used together, allow the processor to address a single peripheral on the bus for the purposes of control and data transfer. A block diagram of the I/O bus is shown in Figure 1.

The lines in the I/O bus carry either levels or pulses. The standard voltage levels of these signals are 0 volts and -3 volts. In most cases a -3 volt signal is applied to a line by a clamped load (which is described in the next section of this chapter), and a 0 volt signal is applied to the line by pulling the line to ground through the collector-emitter junction of a transistor. Pulses on the line have standard durations of 100 and 400 nanoseconds and may be either positive going (-3 volts to 0 volts and back to -3 volts) or negative going (0 volts to -3 volts and back to 0 volts). All other signals on the lines are defined as being levels. In most cases where pulses are mentioned either a pulse or a level transition (in the same direction as the leading edge of the specified pulse) will perform equally well.

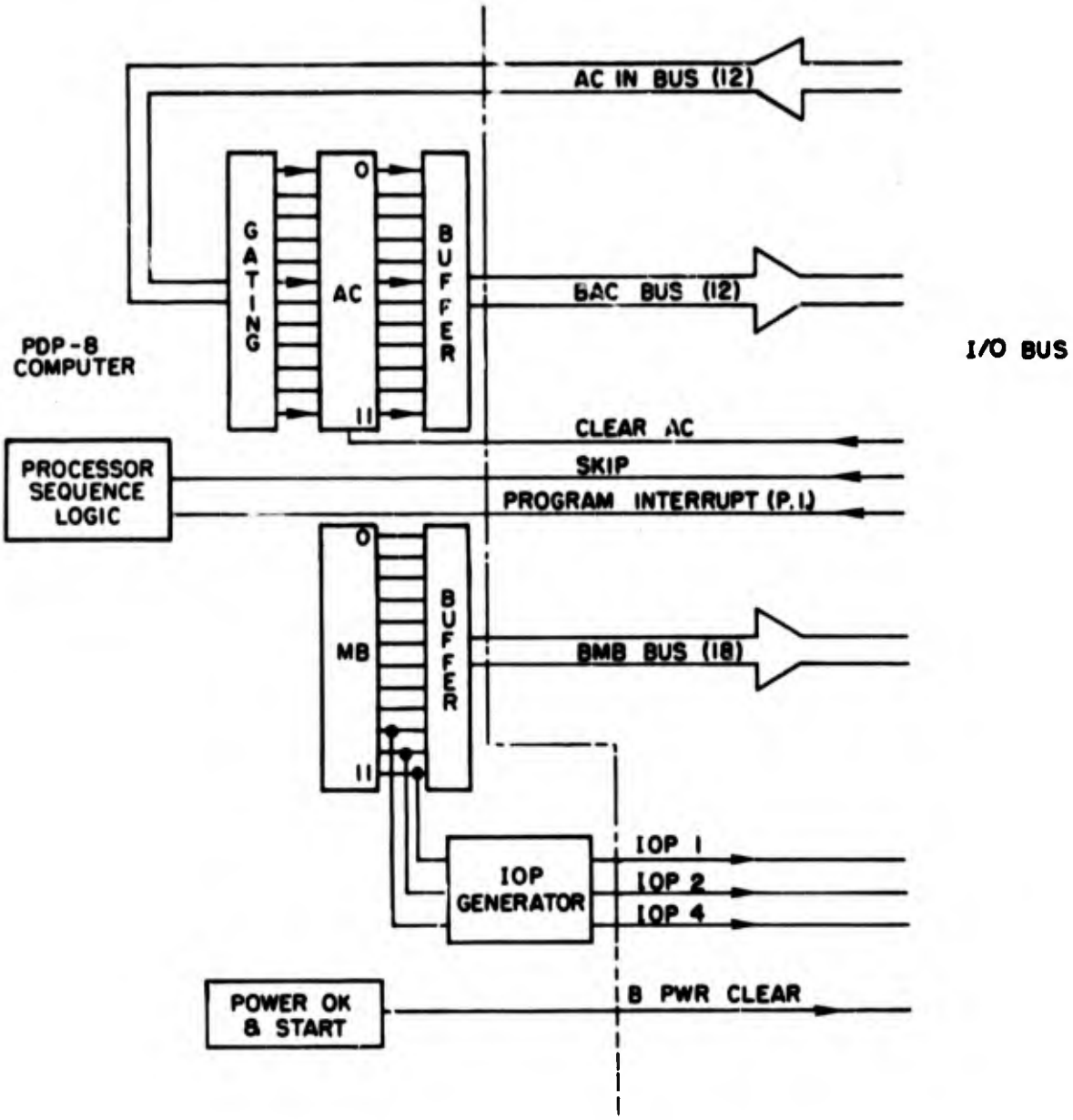


FIGURE 1

PDP-8 PROGRAMMED I/O BUS BLOCK DIAGRAM

The output control lines consist of four pulsed and eighteen level lines. All of these signals except one are determined by the contents of the Memory Buffer (MB) register in the processor. The extra signal line is the B PWR CLEAR or Initialize line. A negative going pulse is put on this line by the processor whenever the power switch on the computer is turned on or the START switch on the computer console is depressed. This signal is used to initialize all device controls, flags, etc. to a known state. The actual effect of the signal depends on the I/O device and its usage is therefore included in the I/O device descriptions.

Of the signals determined by the MB register, the level lines are derived by buffering the outputs of the MB register with amplifiers, and are therefore referred to as the Buffered Memory Buffer (BMB) lines. Generally, only 15 of these 18 lines are used for programmed I/O. These 15 lines are the true and complement outputs of the middle six bits of the 12 bit MB register (12 lines) and the true outputs of the low order three bits of the MB register (3 lines). Although these levels reflect the contents of the MB register during every memory cycle, they are of significance to peripheral devices only during the execution of one family of instructions, the Input/Output Transfer (IOT) instructions (or commands). This family of instructions causes the processor to generate three pulses, called IOP pulses, which are conditioned by the contents of the low order three bits of the MB. At the time these pulses are generated, the MB contains the IOT instruction itself and so the three pulses are actually generated depending on the states of the three low order bits of the instruction. These are the three control pulses mentioned above. They

are labeled IOP1, IOP2 and IOP4 and are related to MB bits 11, 10 and 9, respectively. They are generated in a fixed time sequence during the execution of the IOT instruction, as shown in Figure 2. As this drawing shows, IOP1 is generated only if MB11 (the low order bit) is a 1, IOP2 is generated only if MB10 is a 1 and IOP4 is generated only if MB9 is a 1. It can also be seen that the separation of the pulse generation times is fixed at 1 microsecond. Due to this method of pulse generation the pulses are completely independent from each other (except for their time relationship) and can be used in any desired combination.

Every peripheral device has at least one "Device Selector Module" (which is more fully described in the next section) to which the three IOP pulses and the 12 true and complement lines from the middle six bits of the MB are connected. This module is used by the device to determine when it is being addressed, and at that time to pass the amplified and shaped IOP pulses (now called IOT pulses) to the device. It is important to understand that the use of these IOP (become IOT) pulses is determined entirely by the peripheral device being addressed, not by the processor as is the case with some small computers.

The 12 data output lines are called the Buffered AC lines because they are the buffered outputs of the bits in the Accumulator (AC) register in the processor. These lines can be sampled when it is desired to transfer data from the processor to a device. This is usually carried out in the device by gating these lines with one of the IOT pulses from the device's device selector module. The levels on these lines are 0 volts for bits in the 1 state and -3 volts for bits in the 0 state.

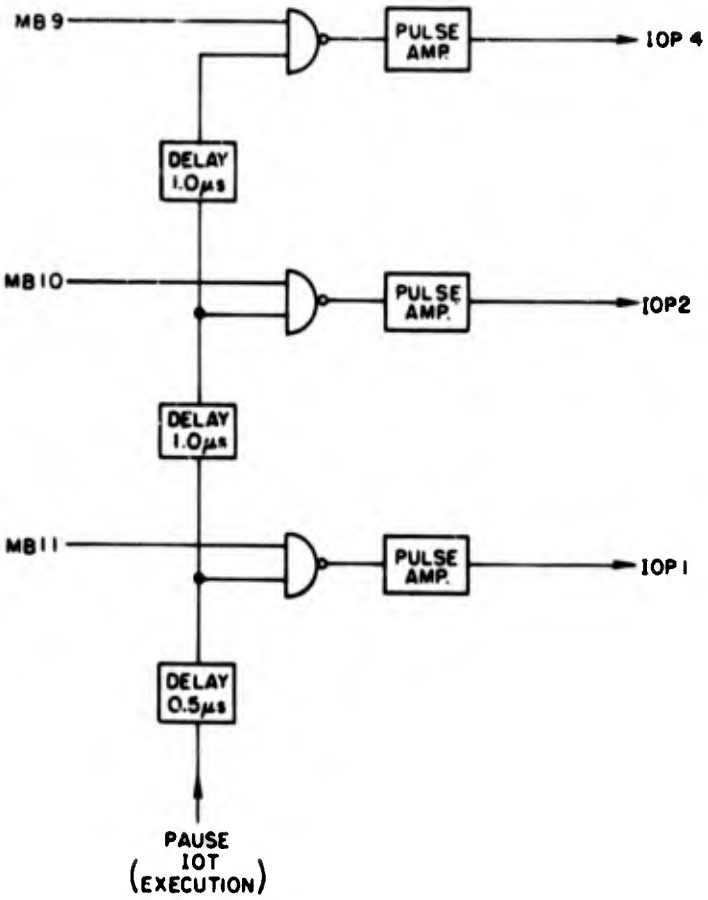


FIGURE 2

PDP-8 IOP GENERATOR LOGIC

The three input control lines are used by peripherals to communicate back to the processor. They consist of two pulsed lines, the AC clear bus and the Skip bus, and one level line, the program interrupt bus. These lines are all held at -3 volts (the 0 state) by clamped loads in the processor, and are pulled to ground (the 1 state) through the collector-emitter junction of a transistor.

The AC clear bus is used by peripheral devices to force all of the bits of the AC register to the 0 state. It may be grounded before data is transferred into the AC of the processor or after data is transferred out of the AC of the processor, but must always be done under the control of a specific IOT pulse from a device selector. This is necessary due to the fact that peripheral devices and the processor are by nature asynchronous and that clearing of the AC at a random time would have disastrous results.

The Skip bus is used to cause the Program Counter (PC) register in the processor to be incremented by an extra 1 at the end of the IOT instruction. The PC is normally incremented by 1 to cause the sequential execution of instructions in memory. Incrementing it by 1 extra (a total of 2) will cause the processor to "skip" the next sequential instruction. This provides a method whereby a peripheral device can respond to an inquiry from the processor concerning the status of some condition(s) in the device. The program in the computer can then take different actions depending on whether or not the skip occurred. As mentioned above for the AC clear bus, this bus must be grounded only in response to a specific IOT instruction from the processor.

The program interrupt (PI) bus is used by peripheral devices to indicate to the processor that they are in need of service. The bus is pulled to ground by any and all peripherals that need service as soon as that service is needed and is held at ground level until the requested service is received. In order for the processor to see the signal on this line, it must be enabled for program interrupts, otherwise it will ignore the signal. If the processor is enabled for interrupts and this line is pulled to ground, it goes through a fixed sequence of operations at the end of the execution of the instruction in progress. This sequence is more fully described in Chapter 3 and Appendix 4 of the Digital Equipment Corporation handbook.[1] Briefly, the contents of the PC is stored in location 0 of memory, the processor is disabled from further interrupts, and execution of the instruction in location 1 of memory begins. The program beginning at this location usually saves the contents of the active registers in the processor (AC, Link, etc.), identifies and services the requesting device, restores the active registers and returns to the interrupted program with the processor again enabled for interrupts. This program interrupt facility is particularly useful when it is desired to operate low- and medium-speed devices at their maximum rate at the same time that the processor is doing other things.

The 12 data input lines are used to strobe data into the AC register in the processor. These lines are held at -3 volts by clamped loads in the processor and are pulled to ground by a device when it is desired to transfer data from it to the processor. This transfer is accomplished by gating bits that should be set to the 1 state with an IOT pulse. These

lines will then be pulled to ground by the collector-emitter junction of a transistor, loading specified bits in the 1 state into the AC. Note that the AC will then contain the inclusive OR of the previous contents of the AC and the 1 bits on the data lines. Since the bits in the 0 state will not affect the bus, the corresponding bits in the AC will not be changed. Where it is desired to load the AC with data from the peripheral, the AC should be cleared first either by an instruction in the processor or by the use of the AC clear bus.

Further descriptions of the details of the characteristics and use of the programmed I/O bus can be found in the Digital Equipment Handbook[1]; however, this section should have provided enough information for understanding the next chapter.

2. Flip-Chip Module Characteristics

As mentioned previously, Flip-Chip modules are discrete component diode-transistor logic modules. The standard logic levels for these modules are 0 volts (ground) and -3 volts. Included in the line of modules are inverters, multiple input NAND/NOR gates, flip-flops and special logic functions. These modules are used both in the devices described in the next chapter and in the PDP-8 computer and its associated peripherals which are produced by Digital Equipment Corporation. This section will simply describe a few of the characteristics of these modules which make them different from the integrated circuit logic used commonly today. A more complete description of these modules can be found in another Digital Equipment handbook.[2]

Following immediately is a description of a "typical" gate,

including the clamped load mentioned earlier, which is representative of the inverters, gates and some of the other functions in the line of modules. Next is a description of the Diode-Capacitor-Diode gate which is an important part of the flip-flop and delay modules. Finally, a description is given of the "Device Selector Module" which is used by all peripherals to determine when they are being addressed.

The "typical" gate.

A multiple-input diode-transistor gate is shown in Figure 3. When all inputs are at -3 volts, the point is also negative and current flows from the transistor emitter through the biasing diodes and biasing resistor R1 to the -15 volt supply. The transistor will be turned on, and will appear as a low collector to emitter impedance. When any one of the inputs goes to ground, the node point will also be pulled to ground. The other input diodes will be back-biased and will thus isolate the other inputs from the node point. Current will now flow through the biasing diodes and biasing resistor R2 and will hold the base of the transistor more positive than the emitter. This will turn off the transistor and it will appear as a high collector to emitter impedance.

The circuit shown in the dash box is the clamped load which was mentioned earlier. When a high impedance (or an open circuit) is connected between its terminal and ground, the voltage at the terminal will go in the negative direction toward the -15 volt supply. When it gets to -3 volts, the clamp diode will conduct and current will flow through it to the -3 volt supply. This will hold the terminal at -3 volts

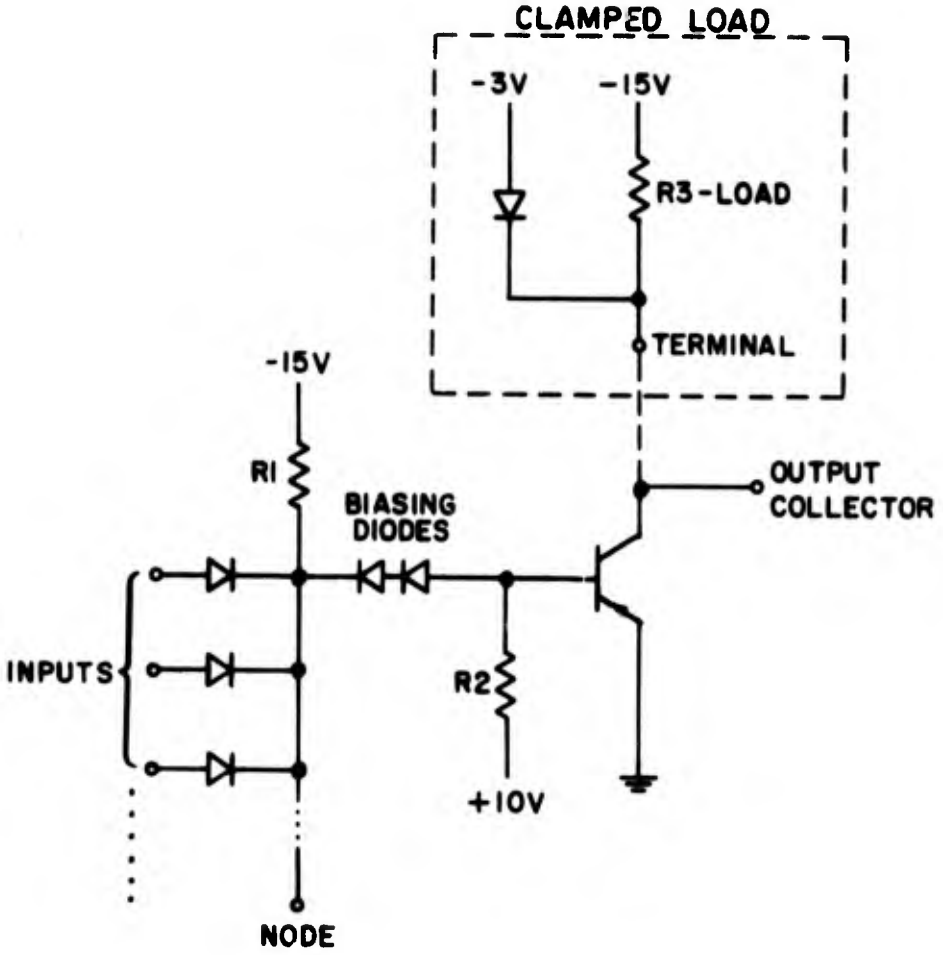


FIGURE 3

MULTIPLE-INPUT DIODE-TRANSISTOR GATE

(with a 12 volt drop across load resistor R3). When a low impedance (or a short circuit) is connected between the terminal and ground, the voltage will go below -3 volts (which will cause the clamp diode to stop conducting), and current will flow from the -15 volt supply through the low impedance (short) to ground. This circuit will thus establish levels of 0 volts and -3 volts depending on whether a low- or high- impedance (respectively) is connected to the terminal.

For standard gates, these two circuits are put on the same module card and work together to carry out the gating function. When used with the I/O busses previously described, the clamped load appears in the processor and the output collectors of the gates in each peripheral device are all tied in parallel on the bus. In both cases, the clamped loads supply the -3 volt level and the collector(s) pull the bus to 0 volts.

Taking the clamped load and the rest of the circuit together, it can be seen that when all inputs are at -3 volts the output will be at 0 volts and when one or more of the inputs is at 0 volts the output will be at -3 volts. This function, depending on which level is chosen as logic "1", is the NAND/NOR function.

The Diode-Capacitor-Diode gate

The Diode-Capacitor-Diode (DCD) gate is shown in Figure 4. This gate is used for input gating for the flip-flops, delays and pulsers in this line of modules. This simple circuit acts as an AND gate and provides logical isolation and delay (which is important for sampling flip-flops at the same time that they are being changed). The level input enables the gate when at 0 volts for at least 400 nanoseconds before a pulse occurs at the pulse input. The pulse input is activated by the

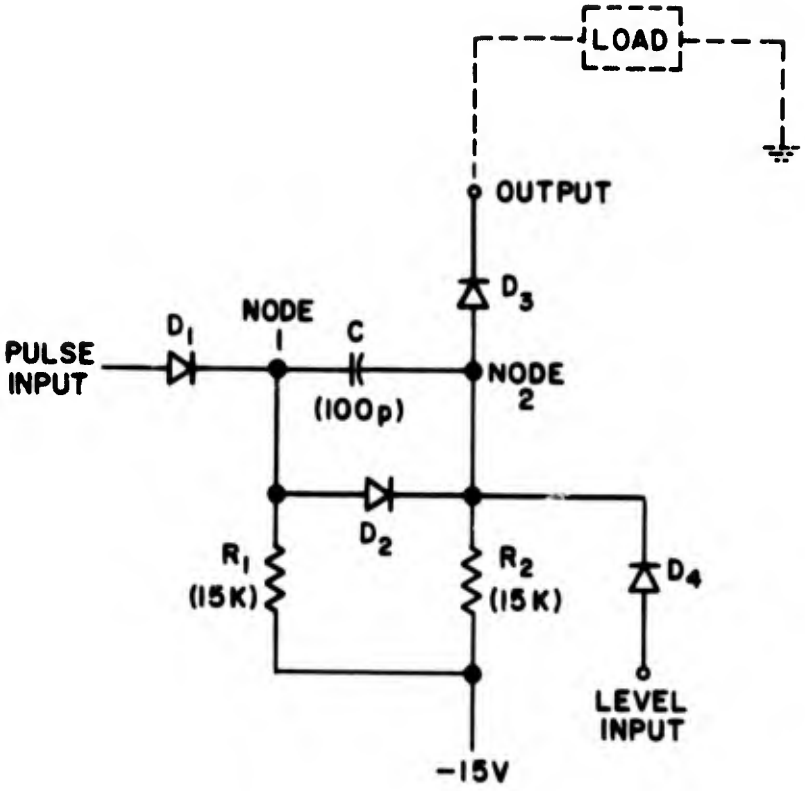


FIGURE 4

DIODE-CAPACITOR-DIODE GATE

positive edge of a level change, and will work equally well with either positive-going pulses or positive-going level changes.

It can be shown that the output terminal, when loaded, will stay at 0 volts as long as the level input is at -3 volts, independent of the condition of the pulse input.

When the level input goes to 0 volts (enabled) while the pulse input is at -3 volts, capacitor C will charge through resistor R1 toward -15 volts. When the voltage across C reaches -3 volts, diode D1 will conduct and clamp node 1 at -3 volts and the capacitor will not charge farther. The time for the capacitor to charge is about 400 nanoseconds. When a positive-going level change appears at the pulse input, diode D1 will cause node 1 to follow the change. This will cause node 2 to go in the positive direction from ground due to the charge on the capacitor. Diode D4 will then turn off and diode D3 will conduct, and the output terminal will go positive. Capacitor C will then discharge through resistor R2. The signal at the output will thus be a positive going pulse. Note that the pulse will appear at the output when the pulse input goes to ground after about 400 nanoseconds from the time that the level input goes to ground. It can also be shown that a pulse will appear at the output if a level change appears at the pulse input within 400 nanoseconds from the time that the level input goes back to the -3 volt state from the 0 volt state (assuming that it was in the 0 volt state for more than 400 nanoseconds).

From the description above, it can be seen that the circuit acts like an AND gate with memory. Although this gate is not shown explicitly

in the equipment schematics in the following chapter, it is the input gate for all of the flip-flops shown, and is particularly important for the proper operation of the variable clock.

The Type W103 Device Selector Module

As described in the first section of this chapter, each peripheral device connected to the PDP-8 computer is responsible for recognizing when it is being addressed. The Device Selector (DS) module is used by peripherals for this purpose. It decodes the middle six bits from the MB register, which is the device address or select code. When it recognizes the proper code, it produces IOT pulses for the use of the peripheral from the IOP pulses that appear on the I/O bus. Although the MB register may appear to have the proper code at times during the execution of instructions other than IOT instructions (since it contains the contents of the currently addressed word for every memory cycle), the IOP pulses are generated by the processor only during the execution of IOT instructions and therefore the output IOT pulses will appear only during an IOT instruction with a device select code identical to the one specified by the module.

The logic diagram for this module is shown in Figure 5. In the schematics in the following chapter it will be shown simply as a logic function. The lower part of the circuit decodes the device select code from the BMB lines. It is simply a 12 input NAND gate, of which only 6 inputs are used. Typically, all 12 inputs are connected to the true and complement outputs of the device select code on the BMB lines. The desired device select code is then selected by cutting off the diodes from the false sides

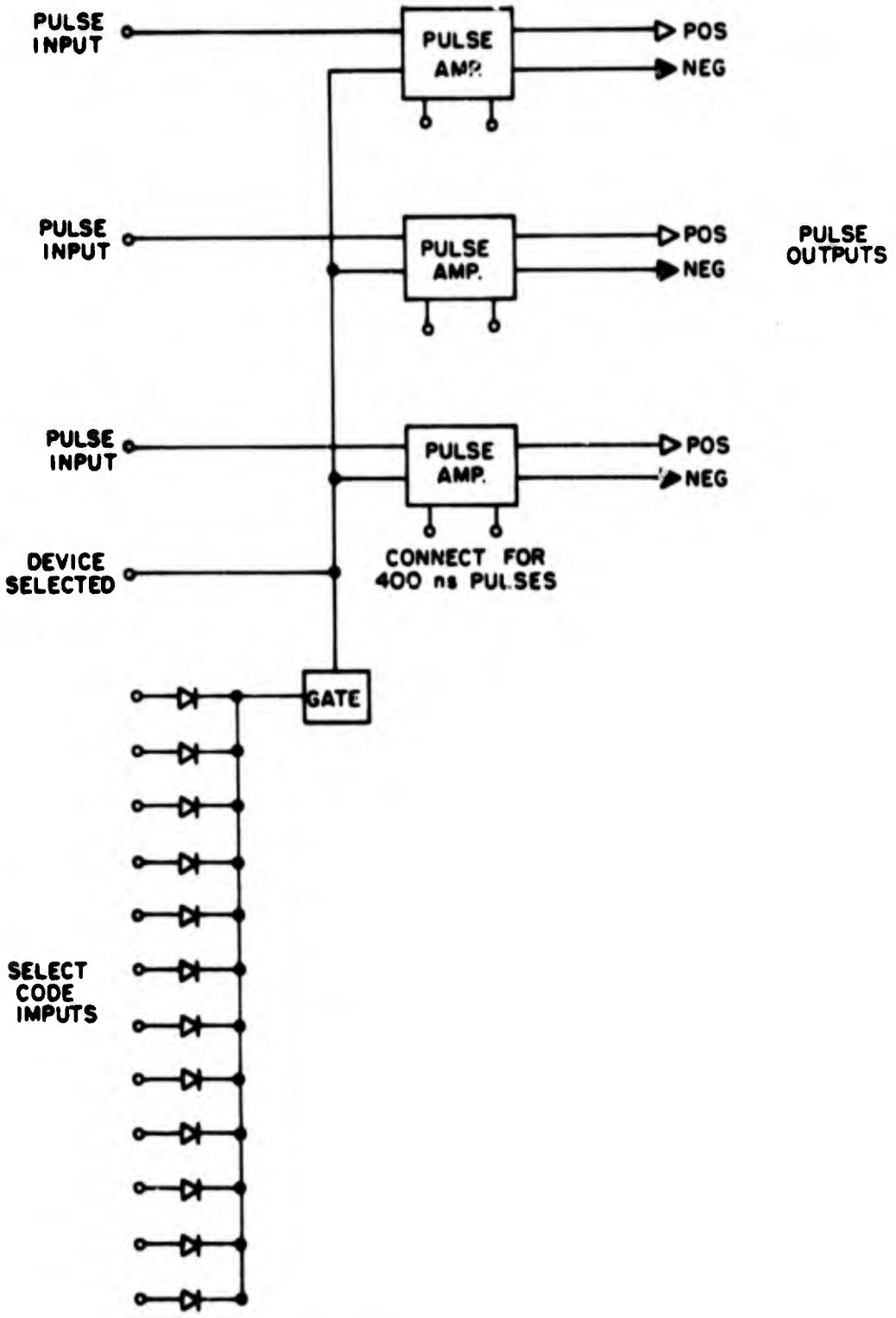


FIGURE 5

TYPE W103 DEVICE SELECTOR MODULE

(opposite of true) of the desired bit combination. The output of the NAND gate will now go to ground when the desired six-bit code appears on the BMB lines.

The output from this select gate is connected to the level inputs of the three pulse amplifiers shown above it. The pulse inputs to these amplifiers are connected to the three IOP lines on the I/O bus. The amplifiers will output a standard pulse of either 100 or 400 nanoseconds (depending on whether or not an external jumper is connected) only when the proper device select code is presented on the BMB lines at least 400 nanoseconds before an IOP pulse appears. The module has thus done most of the work necessary for the presentation of control pulses to the peripheral device, and is usually all that is required for this function.

III. HARDWARE DESCRIPTIONS

This chapter describes the special hardware (non-DEC) devices currently connected to the PDP-8 computer used by the T.P.G. It is intended to be a complete enough description of the hardware to be used as an aid in trouble-shooting or modifying these devices in the future.

The devices described are (1) the Relay Register, (2) the Switch Registers, (3) the Utility Buffer, (4) the 120 Hz Clock, (5) the Variable Clock, (6) the Event Channels, and (7) the Plotter Control and Utility Pulses. Each of the discussions is divided into three or four parts, (a) functional description of the device, (b) commands used for communication with the device, (c) operational descriptions of the device logic, and for some devices, (d) design objectives.

Schematics for each of the devices are included with the discussion of the device. The cable lists for control-to-device connection and the schematic for the Clock and Event Channel Control Box are included in the Appendix.

The relationship of these peripheral devices to the rest of the PDP-8 computer system is shown in Figure 6. Each of the major pieces of equipment in the system is shown in this block diagram. The layout of the Stanford-built peripheral devices is shown in more detail in Figure 7, along with the external devices to which they are connected.

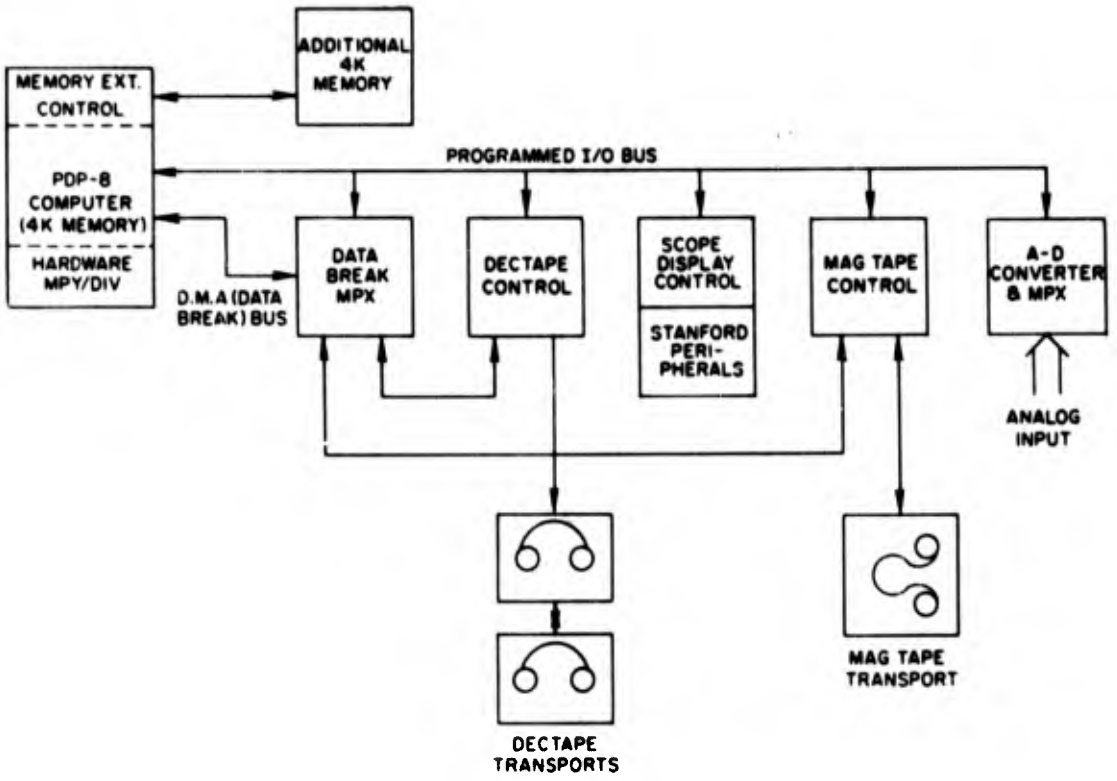


FIGURE 6

SYSTEM BLOCK DIAGRAM

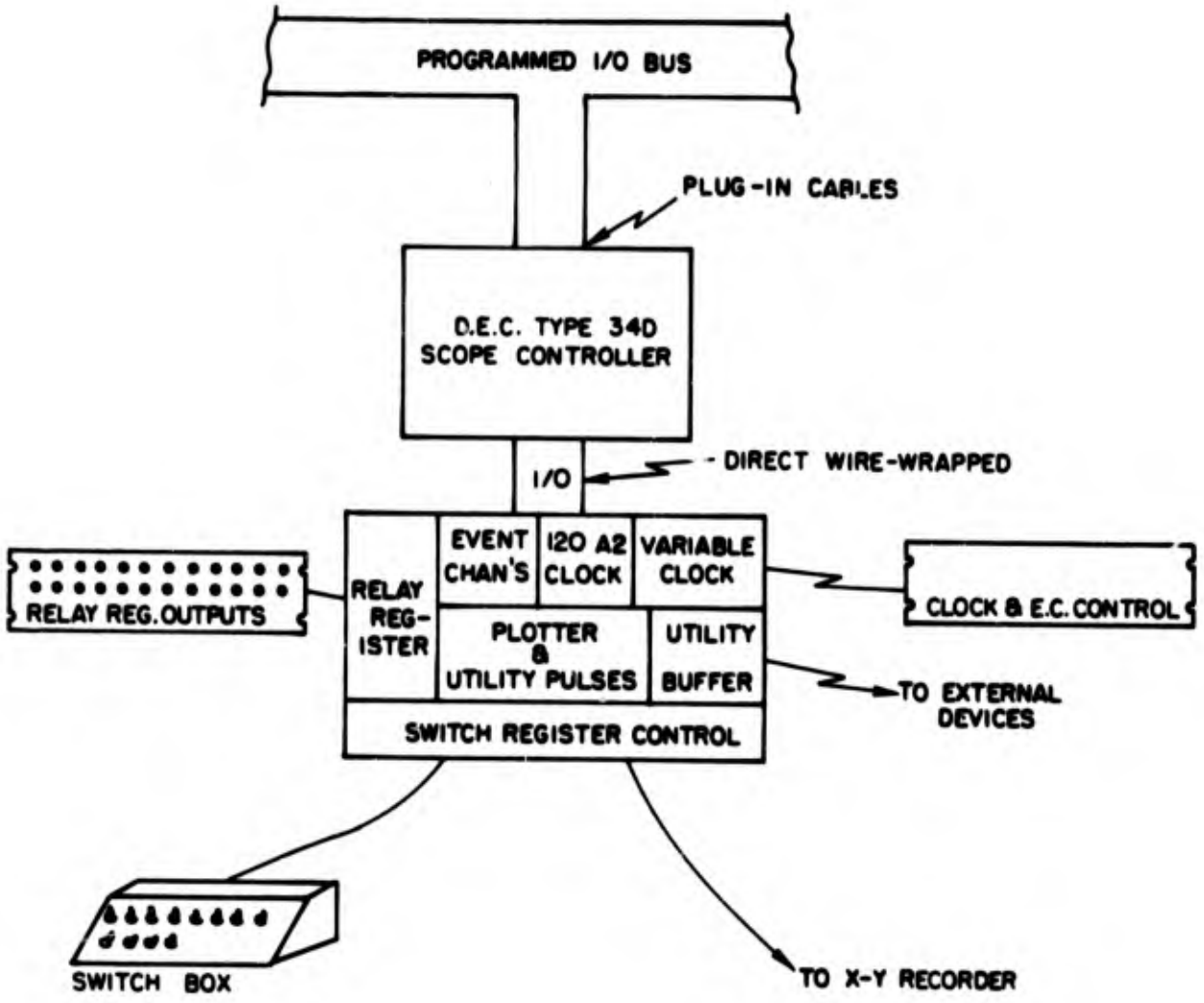


FIGURE 7

BLOCK DIAGRAM OF STANFORD PERIPHERALS

1. Relay Register

Functional Description

The block diagram for the Relay Register is shown in Figure 8. As is shown in this diagram, the device is composed of two major sections, the control section and the register section. The register section is further divided into three parts, a 12-bit flip-flop register with input gating, 12 output gates and 12 output drivers. The design of the device allows any bit in the flip-flop register to be set or cleared independently from the other bits in the register. The flip-flops can also be cleared through the push-button switch-lights connected to the flip-flops. The register can thus be used for both an input and an output device.

The control and register sections of the Relay Register are wired into the logic panel with the other special devices. The push-button switch-lights are mounted on a separate panel on a rack near the computer.

Commands

The IOT commands used by the device all use device code 16 (octal), IOT 616x. The seven commands are as follows:

6161 RARB — Read All Relay Bits. The contents of the Relay Register are OR transferred into the AC.

6162 CSRB — Clear Selected Relay Bits. Flip-flops selected by 1 bits in the AC are set to the 0 state.

6163 CSRB CAC — Clear Selected Relay Bits, Clear AC. Performs the same function as CSRB, and then clears the AC.

6164 SSRB — Set Selected Relay Bits. Bits selected by 1 bits in the AC are set to the 1 state.

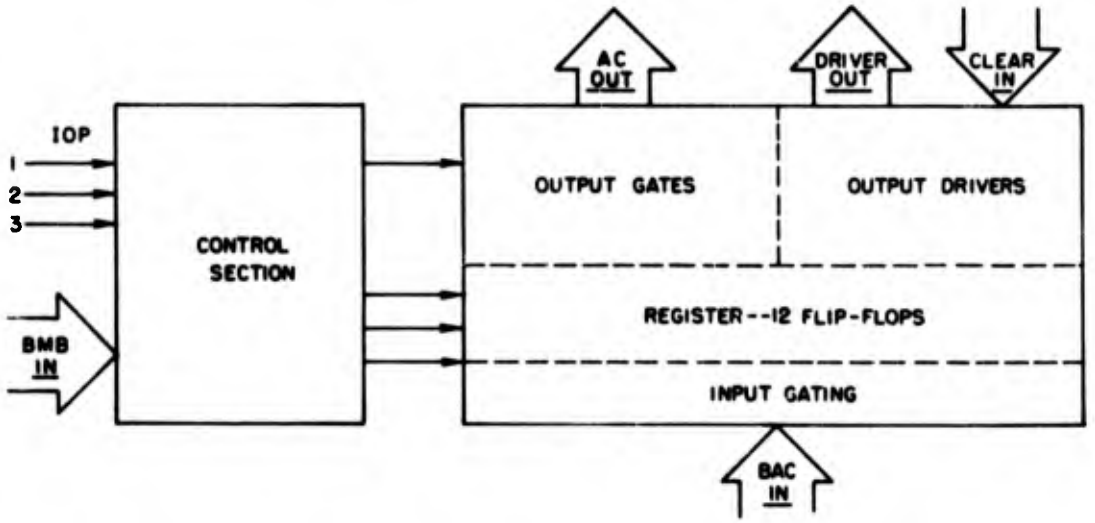


FIGURE 8

RELAY REGISTER BLOCK DIAGRAM

6165 SSRB CAC Set Selected Relay Bits, Clear AC. Performs the same function as SSRB, and then clears the AC.

6166 LARB Load All Relay Bits. Bits in the register are set to the state of corresponding bits in the AC.

6167 LARB CAC Load All Relay Bits, Clear AC. Performs the same function as LARB, and then clears the AC.

Operational Description

The control section schematic is shown in Figure 9. It uses the outputs of BMB bits 9 through 11 (BMB9 through BMB11) to further decode the IOT pulses from the device selector to develop the seven commands. These commands were set up so that all timing information needed for the operation of the register could be derived from the IOT pulses from the device selector module.

There are several important features of the command set which make the decoding of the commands easy. Only one of the seven commands transfers data into the AC. The other six are all output transfers from the AC. Three of these six clear the AC after transferring data from it. Lastly, the last two commands (the load commands) bear some resemblance to the preceding two commands (the set bit commands). Due to these relationships, the decoder needs to derive only five discrete functions from the seven commands. These functions are (1) read, (2) clear selected, (3) set selected, (4) clear all bits, and (5) clear AC.

The read function is derived directly from the read command,

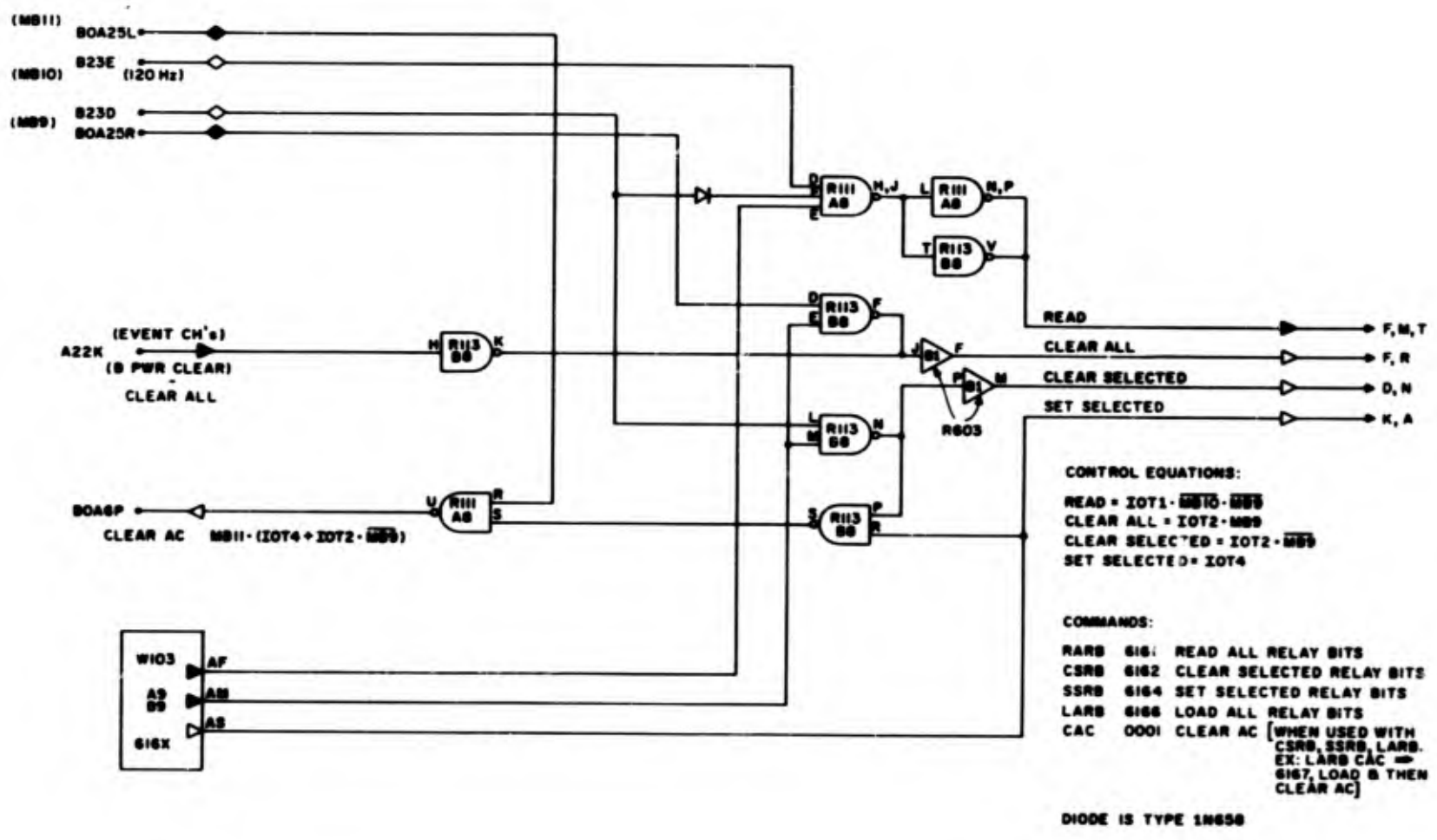


FIGURE 9

RELAY REGISTER-CONTROL SECTION

6161. This function is defined as follows:

$$\text{READ} = \text{IOT1}.\text{and}.\overline{\text{BMB9}}.\text{and}.\overline{\text{BMB10}}$$

In other words, the read function is done at IOT1 time if both BMB9 and BMB10 are not in the 1 state (neither IOT2 nor IOT4 will follow for this command). The implementation of this function is shown in Figure 9. The IOT1 pulse and BMB lines are connected to pins E, D and F (respectively) of the R111 NAND gate at A8. The output of this gate will be at 0 volts only when the equation shown above is true. This output is connected to an R111 and an R113 gate which are connected as inverters. They are connected in parallel to provide enough current drive capability. Their output is the read function.

The clear selected function is derived directly from the clear selected bits commands, 6162 and 6163. This function is defined as follows:

$$\text{CLEAR SELECTED} = .\text{not}.(\text{IOT2}.\text{and}.\overline{\text{BMB9}})$$

In other words, the clear selected function is done at IOT2 time if BMB9 is not in the 1 state (IOT4 will not follow for this command). The function is implemented by a R113 (pins L, M and N at B8) NAND gate and a R603 pulse amplifier (pins P and M at B1). The output of this gate will be at 0 volts when both inputs are at -3 volts, which will satisfy the logic equation above. This is the clear selected function, and is amplified by the pulse amplifier to provide the necessary drive.

The set selected function is derived from both the set selected commands, 6164 and 6165, and the load commands, 6166 and 6167. The

function is defined as follows:

SET SELECTED = .not.IOT4

where the .not. specifies that the function is present when the output of the device selector module is at 0 volts. As can be seen, this function does not require any further gating.

The clear all function is derived from the load commands, 6166 and 6167, and from the B PWR CLEAR line from the I/O bus. This function is defined as follows:

CLEAR ALL = .not.(IOT2.and.BMB9 .or. B PWR CLEAR)

In other words, the clear function is a 0 volt level which is present when either the B PWR CLEAR pulse is sent from the computer, or when IOT2 is present at the same time that BMB9 is in the 1 state (IOT4 will follow for these commands). This function appears at IOT2 time for the load commands, and clears the register so that the set selected function can be used to properly make the contents of the register agree with the contents of the AC. The function is implemented by inverting the B PWR CLEAR pulse with a R113 NAND gate (pins H and K at B8) and using its output in a wired-OR configuration with another R113 NAND (pins D, E and F at B8) whose inputs are the IOT2 pulse and the true side of BMB9. The ORed result is the input to a R603 pulse amplifier (pins J and F at B1), which is needed to provide the necessary drive.

The clear AC function is derived from all of the output transfer commands which end with an odd digit, 6163, 6165 and 6167. The function is defined as follows:

CLEAR AC = .not.BMB11.and.(IOT2.and.$\overline{\text{BMB10}}$ .or. IOT4)

In other words, if the command is odd but not read, clear the AC at IOT2 time unless IOT4 is to follow, in which case clear the AC at IOT4 time. It is necessary that the function not be true until the last IOT pulse for the command, or else the AC might be cleared at IOT2 time during the execution of the load command 6167, in which case the register would not be set as specified by the 1 bits in the AC. The function is implemented by NORing the clear selected function (which cannot occur if IOT4 is to follow) with the IOT4 pulse from the device selector using a R113 gate (pins P, R and S at B8). The output of this gate is NANDed by a R111 (pins R, S and U at A8) with BMB11 and connected to the CLEAR AC line in the I/O bus. The output from this gate will pull the I/O bus line to ground when the function is true.

The register section is shown in Figure 10. The function lines from the control section (described above) are shown entering the drawing from the left and are bussed to all of the flip-flops in the register. Since all bits of the register are functionally the same, only bit 11 (upper right in the drawing) will be described.

The level inputs to the R202 flip-flop (pins P and V at B7) are both connected to the BAC bus. They will be held at 0 volts when bit 11 of the AC is in the 1 state. The set selected and clear selected function lines are connected to the set and clear pulse inputs, respectively. When one of these two pulses appears, the flip-flop will go to the specified state if the level inputs are also at 0 volts (corresponding to a 1 bit in the AC). Thus these two functions are performed completely

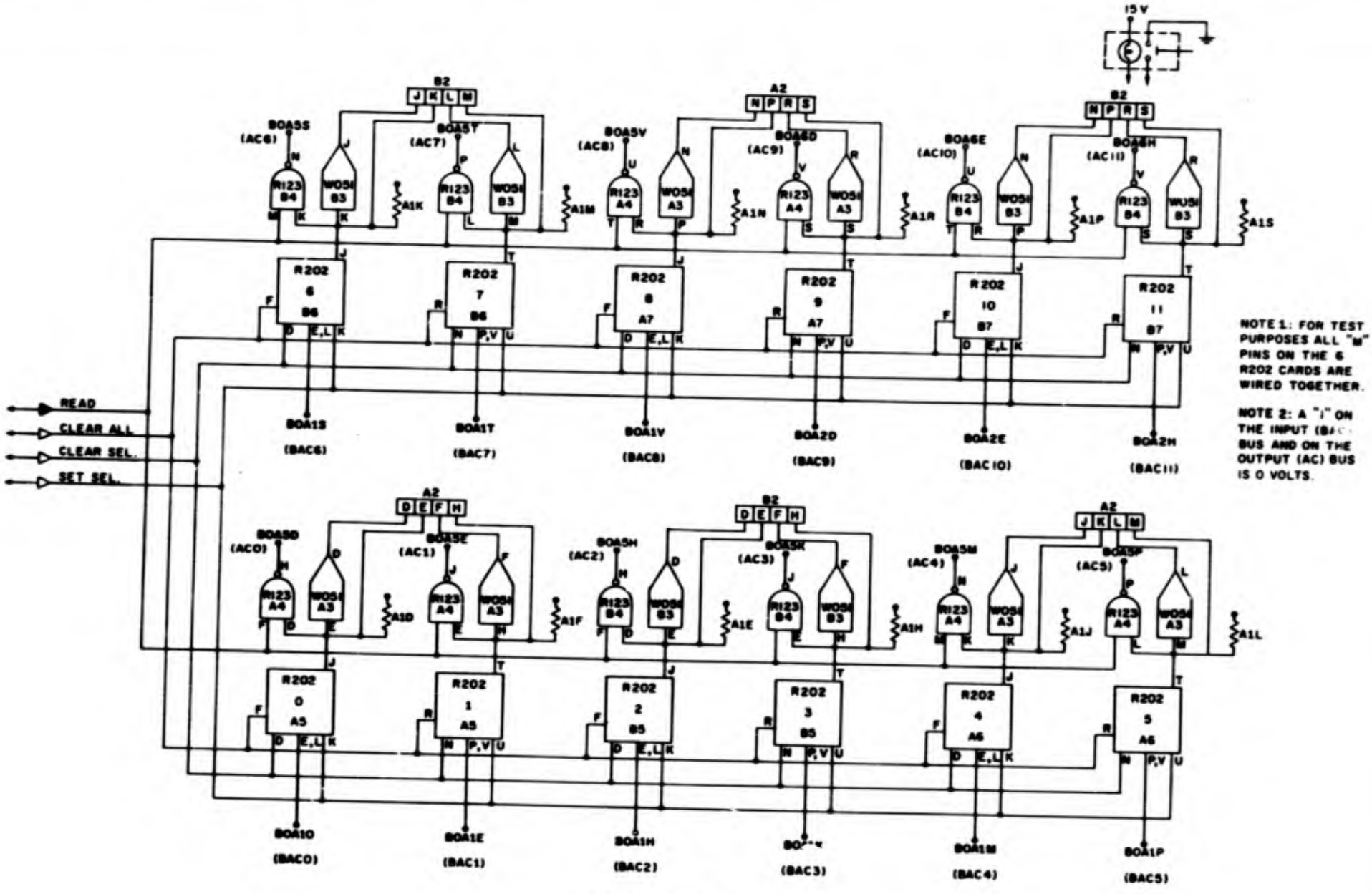


FIGURE 10

RELAY REGISTER-REGISTER SECTION

by the input gating of the flip-flops themselves. The clear all function line is connected to the direct clear input (pin R at B7) of the flip-flop. When it is asserted (0 volts), the flip-flop will be forced to the 0 state.

The "1" output (-3 volts when true) from the flip-flop is connected to several circuits: a R123 output gate, a W051 relay driver, a clamped load, and a flip-flop clear line to the external control box. The output gate is used by the read function to gate the current state of the flip-flop onto the AC input bus. The gate output is an open collector transistor without a clamped load. If the flip-flop is in the 1 state when the read pulse occurs, the bus will be pulled to 0 volts, which will set the corresponding bit in the AC (an OR transfer).

The relay driver provides the necessary drive for lights and relays. Its output is connected to the light in the push-button light and to a BNC connector (which can be used to drive a relay or another light) in the external control box. The driver is a 100 ma driver and the light draws 30 ma, leaving 70 ma of capacity. Care should be taken to limit the load to 100 ma or else the driver transistor will burn out. If a light or relay is connected to the BNC, the other side of the device must be connected to either the -15 volt BNC which is provided on the external control box or to a good -15 volt power supply (the supply should be connected to the ground side of the computer power supply).

The clamped load is connected to the output of the flip-flop to prevent ringing on the flip-flop clear line from forcing the flip-flop to the 0 state after it has been set to the 1 state. This line, which

extends to the external control box, allows the flip-flop to be cleared (set to the 0 state) by pushing the button on the push-button lights. When the push-button is closed, the output of the flip-flop is tied to ground. If the flip-flop is in the 0 state, its output will already be at 0 volts, and nothing will happen. If the flip-flop is in the 1 state, with its output at -3 volts, forcing the output to 0 volts will force the flip-flop to the 0 state, where it will stay until set again. This feature allows the bits in the register to be used as input switches (by reading the contents of the register to see if any of the bits have changed from the 1 to the 0 state). It also means that bits which should stay set may need to be reset should the button be pushed.

2. Switch Register

Functional Description

The block diagram for the Switch Registers (SRs) is shown in Figure 11. As can be seen from the diagram, the device is composed of a control section and an externally mounted switch box. The control section is contained in part of the mounting panel that contains the X,Y display control. The switch box is mounted under the console of the computer, directly under the console switch register, and contains three 12-bit switch registers, SR1, SR2, and SR3. The two sections are connected by a cable.

The control section does a small amount of decoding (beyond that done by the device selector module) to select the one of the three

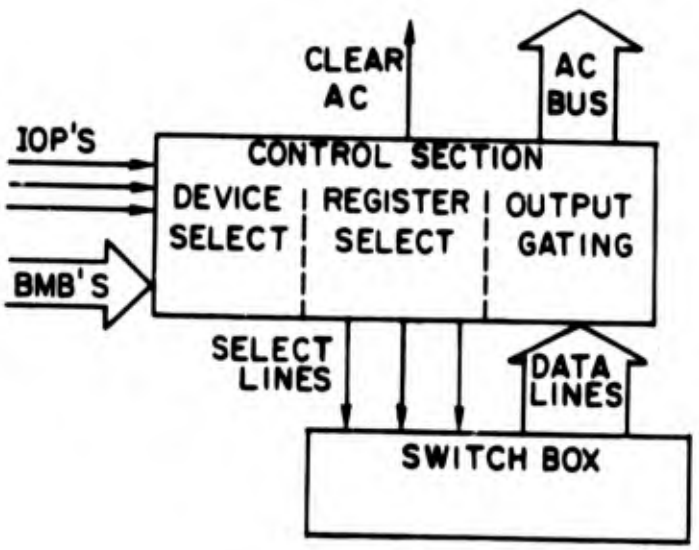


FIGURE 11

SWITCH REGISTERS - BLOCK DIAGRAM

switch registers to be read and, if desired, to clear the AC.

Commands

The IOT commands used by the device all use device code 15 (octal), IOT 615x. The six commands are as follows:

6152 OSR1	OR Switch Register 1 with the AC.
6153 LAS1	Load the AC with Switch Register 1.
6154 OSR2	OR Switch Register 2 with the AC.
6155 LAS2	Load the AC with Switch Register 2.
6156 OSR3	OR Switch Register 3 with the AC.
6157 LAS3	Load the AC with Switch Register 3.

Operational Description

The control section schematic is shown in Figure 12. The logic decodes BMB bits 9 and 10 (BMB9 and BMB10) to select one of the three switch registers. This is done by three R111 triple-input NAND gates (at BOD20). These gates are also connected to the "device selected" output from the device selector module through a R111 NAND gate (pins R, U and V at BOD21) connected as an inverter. These inputs enable the gates to decode BMB9 and BMB10 only when the device is selected. The other inputs to the gates are the true or complement values of BMB9 and BMB10 chosen such that the output of just one of the gates will be at 0 volts when the decoding is done. The connections decode BMB9 and BMB10 for IOT 615x as follows: for 01 (6152 or 6153), pin H is low, selecting SR1; for 10 (6154 or 6155), pin N is low, selecting SR2; for 11 (6156 or 6157), pin U is low, selecting SR3. These output pins are connected to the three output lines to the switch registers.

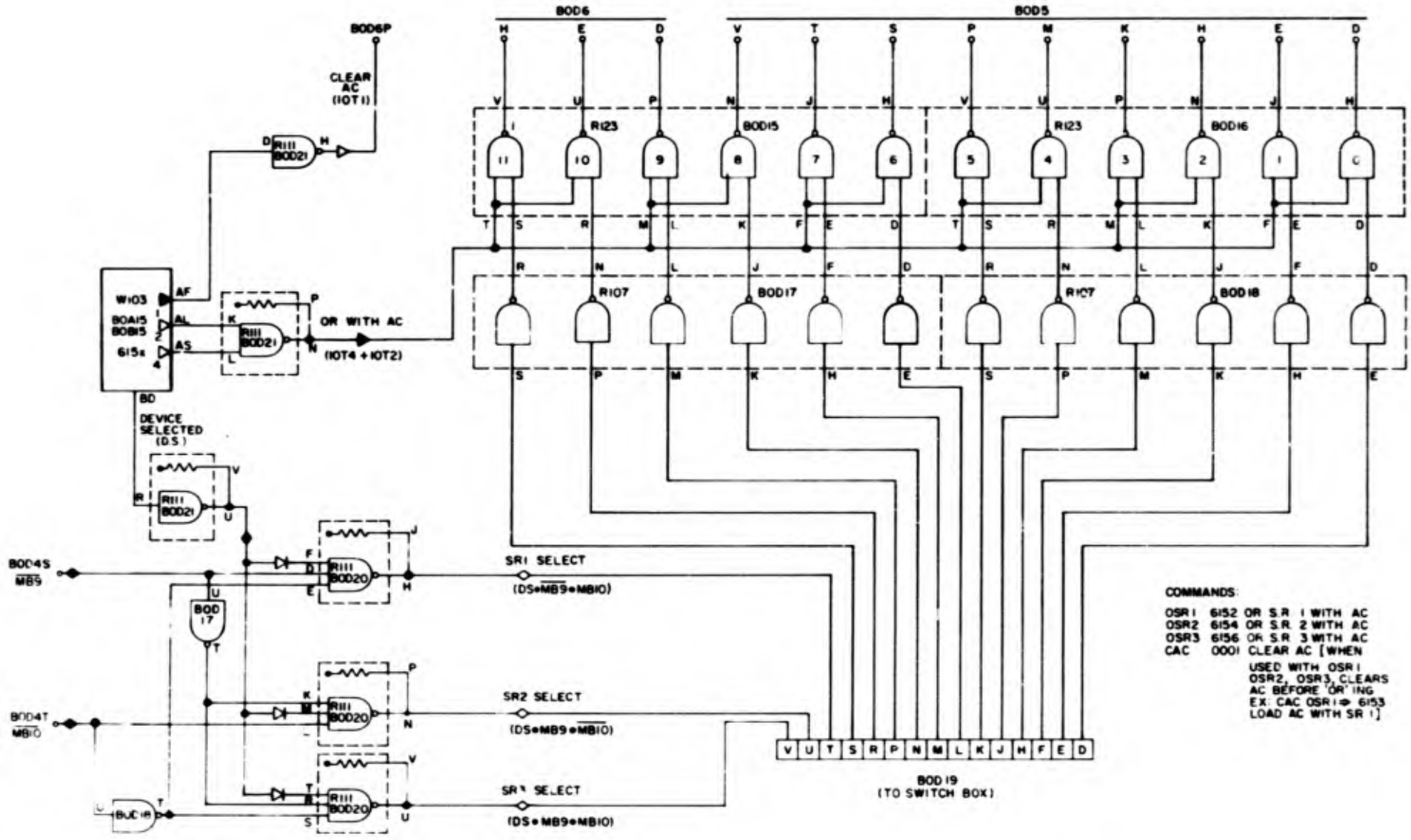


FIGURE 12

SWITCH REGISTERS - CONTROL SECTION

The IOT1 output pulse from the device selector module is simply connected to the AC Clear line through an input bus gate, the R111 NAND (pins D and H) without a clamped load at BOD21. This will cause the AC to be cleared at IOP1 time whenever the 615x IOT command specified ends with an odd octal digit, and causes the LASn commands to effect a jam transfer into the AC by first clearing it and then doing an OR transfer with the set switches of the selected switch register.

The switch register select lines are bussed to all of the switches in each of the 12-bit registers, as shown in Figure 13. The switches are connected in a simple matrix with the three select lines for inputs and the 12 bit lines for outputs. Each of the 36 switches in the three registers is connected to its bit line through a diode. These diodes provide isolation between the three registers. When one of the registers is selected, its select line is pulled to ground (0 volts). Due to the direction that the diodes are connected, any switches in the selected register which are in the 1 state will pull their bit lines to ground also. The diodes keep the grounded bit lines from feeding back through any other switches on the same line which might be in the 1 state to the unselected register select lines.

The bit lines from the switch box then go back to the control section as shown in Figure 12. They are connected through the R107 inverters (at BOD17 and BOD18) so that switches which are in the 1 state will present -3 volts to the inputs to the R123 Input Bus Gates (at BOD15 and BOD16). The data is then gated onto the AC input bus by the "OR with AC" pulse. This pulse is developed by NORing (with the R11x, pins K, L, N and

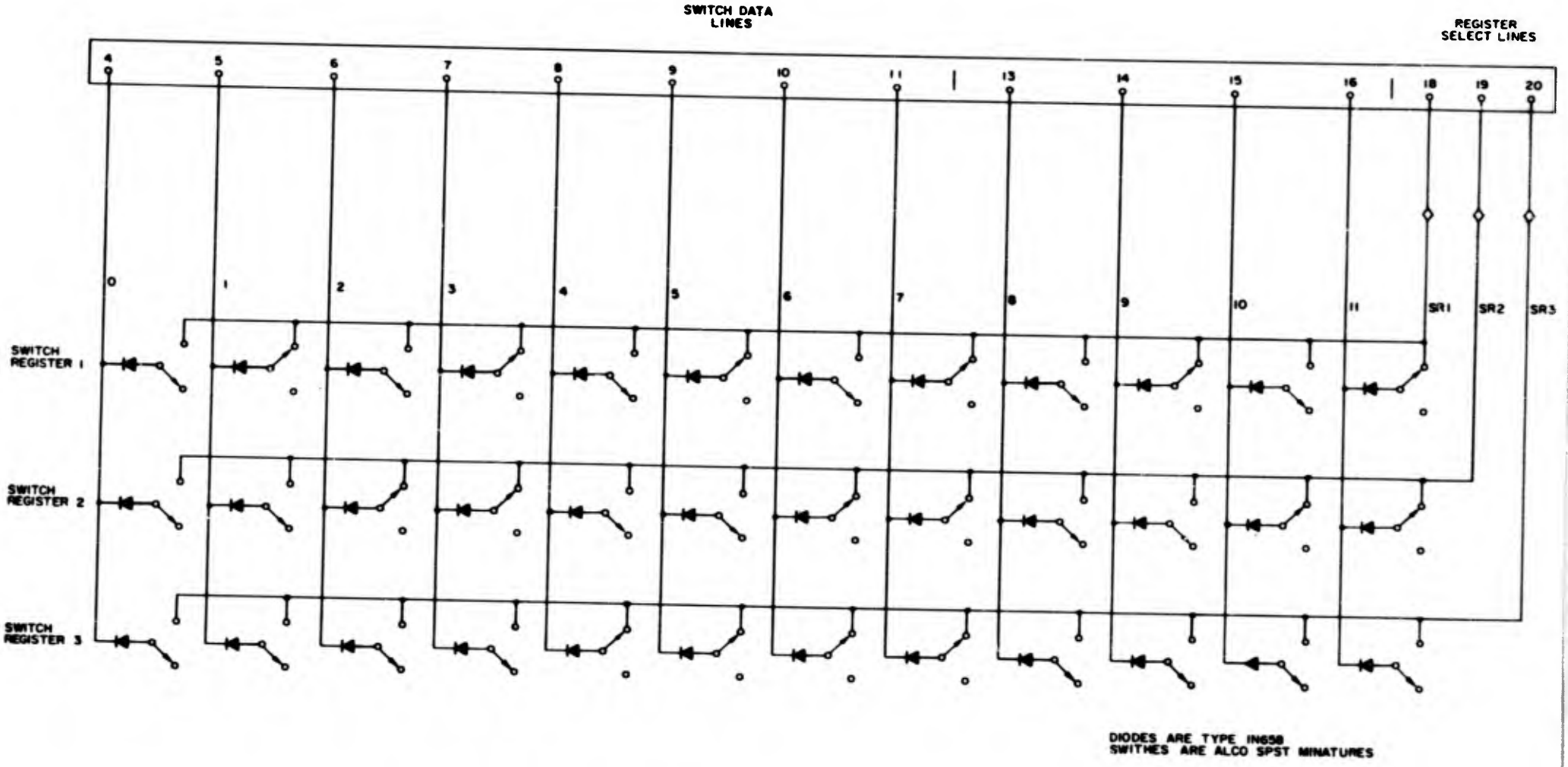


FIGURE 13

SWITCH REGISTERS - SWITCH BOX

P at BOD21) the IOT2 and IOT4 pulse outputs from the device selector. Although this simple method will cause the data to be transferred twice when SR3 is selected, the transfer is an OR transfer and thus will be the same if done twice.

Finally, it should be noted that the arrangement of the control section causes the select lines to the specified register to be pulled to ground for at least 2 microseconds before the data is gated onto the AC input bus. This gives ample time for the select and data lines to settle before the data is transferred through the input bus gates onto the input lines.

3. Utility Buffer

Functional Description

The block diagram for the Utility Buffer (UB) is shown in Figure 14. The buffer consists of input gating, a four-bit up-counter, level converters and light drivers, and control logic. The buffer can be set from the BAC bus and can be incremented either by an IOT or by an external pulse from a push-button or signal generator. The level converters interface the counter with external logic. The light drivers are connected to the Clock and Event Channel Control Box, which is mounted in a rack near the computer. The lights are push-button types, the buttons of which can be used to clear the flip-flops in the counter (when so enabled by a switch on the control box).

Commands

The IOT commands used by the device all use device code 11 (octal),

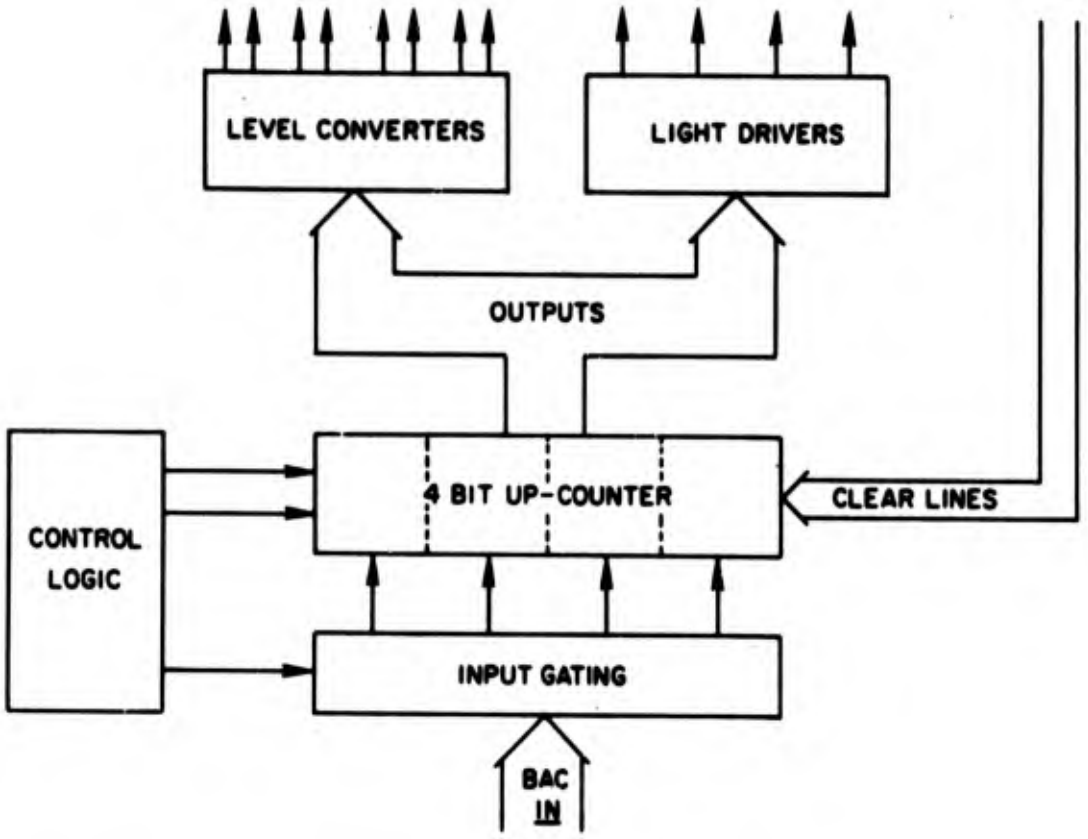


FIGURE 14

UTILITY BUFFER BLOCK DIAGRAM

IOT 611x. The three commands are as follows:

6111 CUB Clear Utility Buffer.

6112 SUB Set Utility Buffer to $\overline{\text{AC8-11}}$.

6113 IUB Increment Utility Buffer.

Normally CUB and SUB are used together to first clear and then set the contents of the buffer.

Operational Description

The schematic for the Utility Buffer is shown in Figure 15. The input gating is accomplished by four of the R123 NAND gates at B28. The gating setup causes the transfer of data to be equivalent to an OR transfer with the complement of AC bits 8 through 11, i.e., bits that are in the 0 state will cause corresponding flip-flops to be set, but bits in the 1 state will not affect the corresponding flip-flops. Normally the flip-flops are first cleared by the CUB IOT and then set by the SUB IOT (as specified by 0 bits in the AC).

The up-counter is composed of four flip-flops which are connected to complement when the output of the previous stage goes from the 1 to the 0 state. The direct clear inputs to the flip-flops are also connected to the IOT1 (CUB) pulse from the device selector module. When an increment pulse is generated by one of the three sources, it causes the first flip-flop ("1") to complement. If it was in the 1 state, the transition to the 0 state will cause the second flip-flop ("2") to complement. This will continue for each of the flip-flops that were in the 1 state before the pulse was generated.

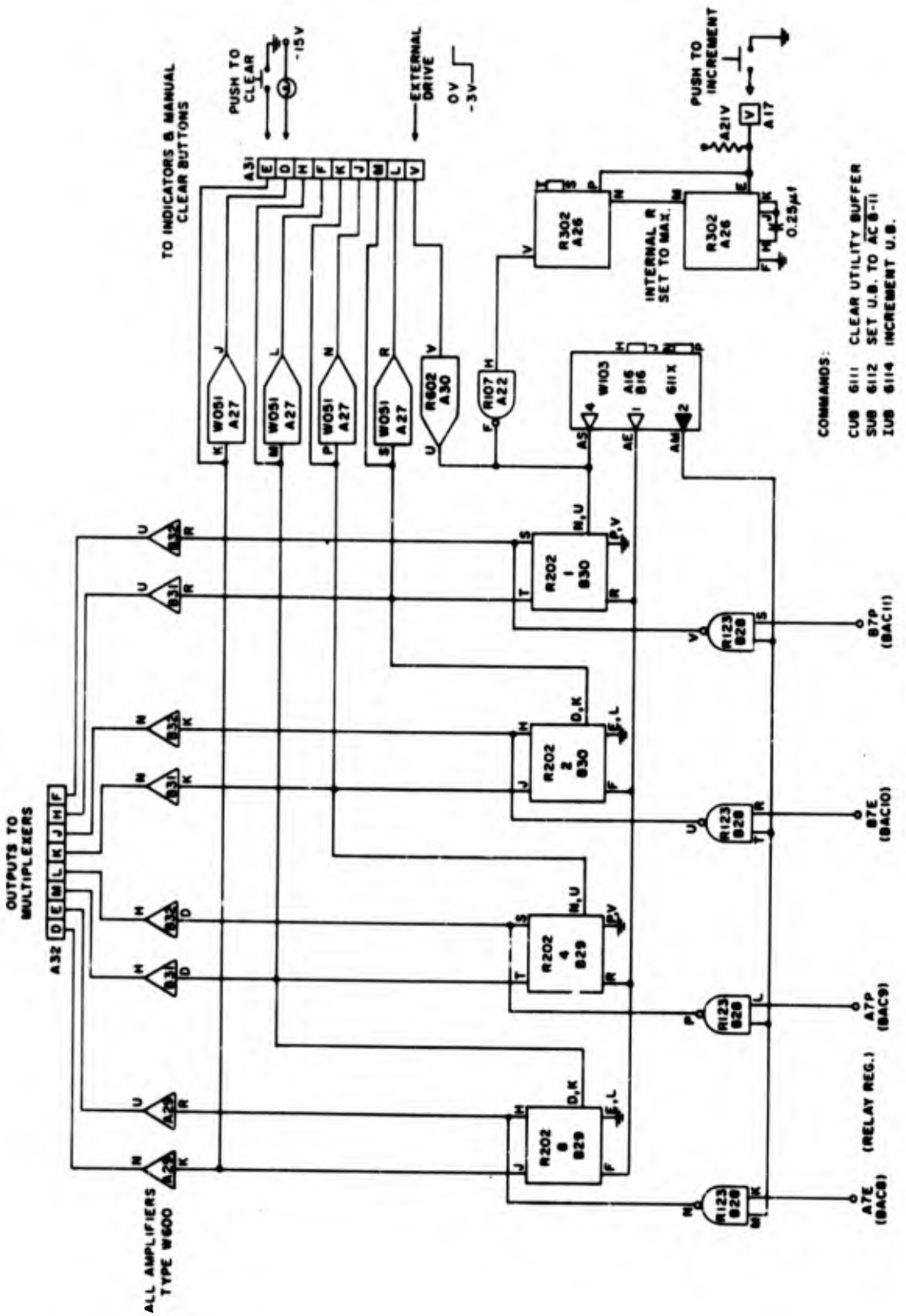


FIGURE 15

UTILITY BUFFER (MULTIPLEXER DRIVER)

The level converters provide both conversion of the 0 and -3 volt outputs from the counter to -15 and 0 volt levels (inversion) and buffering (isolation) of the counter outputs from the output lines. They may be used to provide negative outputs of from -1 to -15 volts, but are currently allowed to output -15 volts due to the fact that the clamp inputs are not connected to anything.

The relay/light drivers simply drive four indicator lights to show the current status of the counter. The lights, which are push-button lights, are also wired to the 1 output side of each of the flip-flops in the counter, and can be used to reset the flip-flops in the counter manually. When a button is pushed, it forces the 1 output of the corresponding flip-flop to 0 volts, which will force the flip-flop to the 0 state if that is not its current state. There is one other result of pushing a button. The 1 to 0 transition, if there is one, of the flip-flop will also cause complementation of the next flip-flop in the counter, in the same way that the increment pulses work.

The control logic consists of a device selector module and two pulse sources. An R602 Pulse Amplifier (pins U and V at A30) isolates the counter from the external drive input and provides pulse standardization for the input signal. The other pulse source is the two connected sections of the R302 Delay (at A26), which is used to provide bounce filtering and pulse standardization for the increment push-button. The pulse input for the lower section on the drawing (pin E at A26) is connected to the push-button and to a clamped load. The clamped load keeps the input at -3 volts until the button is pushed. The transition from -3 to 0 volts triggers the first delay, which has a delay of about

20 milliseconds to allow for the bouncing of the switch contacts. The output of the first delay is connected to the pulse input to the second section (pin N at A26). The level input to this second delay (a DCD input gate) is also connected to the switch and clamped load. This second delay will only be triggered by the transition from -3 to 0 volts on the output of the first delay (at the end of the output pulse) if the switch is closed. This second delay therefore will only be triggered by the closing of the switch, and will ignore both bounce on closing and bounce on opening. The output of the second delay, a pulse of about 400 ns duration, is then inverted and ORed with the other pulse source and the pulse output from the device selector.

4. 120 Hertz Clock

Functional Description

The block diagram for the 120 Hz clock is shown in Figure 16. The clock consists of a 120 Hz source which is driven from the AC power line, flag and control flip-flops, gating to the Skip and PI busses, light drivers and a device selector module.

The clock is enabled or disabled by the enable flip-flop. When in the disabled state, the flag flip-flop cannot be set by the 120 Hz source. It is thus effectively "turned off." The connect flip-flop connects the flag flip-flop to the PI bus. When in the disconnected state, the flag flip-flop cannot request a program interrupt. The device selector module provides the control functions to the various flip-flops. Lastly, the light drivers provide an indication of the status of the control flip-flops to the outside world. The lights

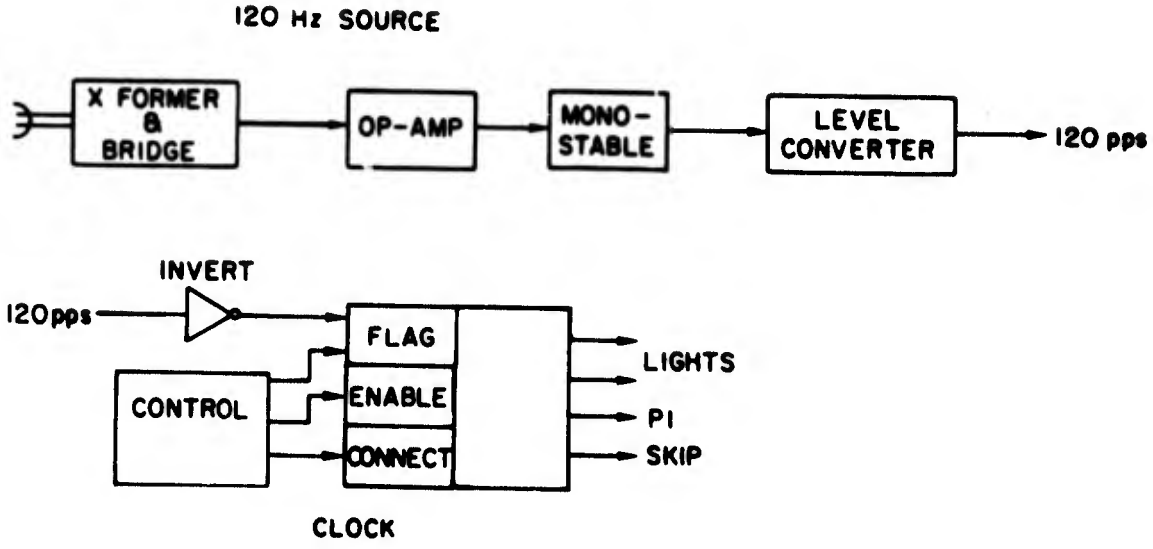


FIGURE 16

120 HZ CLOCK - BLOCK DIAGRAM

that they drive are on the Clock and Event Channel Control box, and are push-button lights which are also connected back to the control flip-flops to allow them to be cleared from the control panel.

Commands

The IOT commands used by the device all use device code 10 (octal), IOT 610x. The four commands are as follows:

6101 TMSF	Skip if Time flag NOT set.
6102 TMCF	Clear Time Flag, connect flag to PI bus and enable Time Clock.
6104 TMDF	Disconnect Time Flag from PI bus.
6105 TMDC	Disable Time Clock and disconnect flag from PI bus.

When it is desired to turn off the clock, TMCF and TMDC are micro-coded together to clear the flag flip-flop and then disable and disconnect the clock. The clock is re-enabled and re-connected by giving TMCF.

Operational Description

The schematic for the 120 Hz source is shown in Figure 17. A transformer with a 6.3 VAC secondary provides voltage at power line frequency to a full-wave bridge rectifier. The 120 Hz output from the rectifier is then applied directly to the input of a μA709 IC operational amplifier which has no feedback. The other input to the operational amplifier is an adjustable, DC threshold voltage. As the input from the rectifier swings past the threshold voltage, the output from the amplifier swings from the maximum negative to the maximum positive voltage that

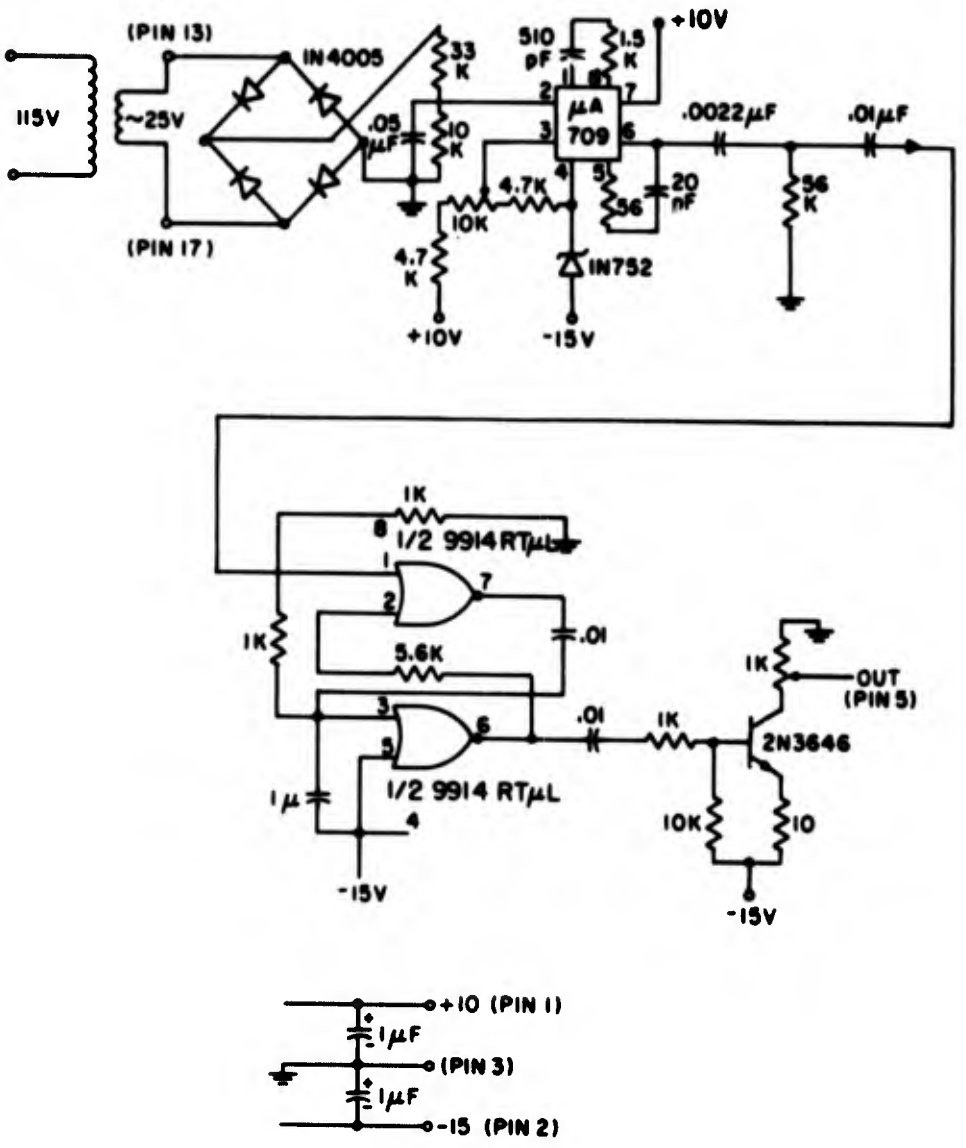


FIGURE 17

120 HZ SOURCE

the power supplies will allow. This output is thus a rectangular wave whose symmetry is adjusted by the threshold voltage. It is connected to the input of a 9914 RTμL dual NAND gate which is connected as a one-shot. The output from this circuit is a pulse of about 10 μs duration at each leading edge of the rectangular wave from the amplifier. The pulse from the one-shot then goes to the final stage, a single transistor, which inverts the pulse and provides level conversion to DEC logic levels. The complete circuit for the source, with the exception of the transformer, is mounted on a Type W994 blank module. The output from the transistor is brought to a pin on this module for connection to the rest of the clock.

The schematic for the 120 Hz clock logic is shown in Figure 18. The output from the 120 Hz source is connected to the input to a R107 inverter (pins R and S at A22), which completes the standardization to DEC signal characteristics. The output from the inverter is connected to the pulse input (pin H) of the flag flip-flop (a R203, pins D, E, F, H and J at B24). The level input (pin J) of the flag is connected to the output from the R203 enable flip-flop (pins K, L, M, N and P at B24). The enable flip-flop, as described previously, is used to turn the clock on and off. When it is in the 1 state (as indicated by the enable status light), the level input to the flag flip-flop will be at 0 volts, and pulses from the 120 Hz source will set the flag. When the enable flip-flop is in the 0 state, the level input to the flag flip-flop will be at -3 volts, and pulses from the 120 Hz source will not affect the flag. In this state the clock is turned off.

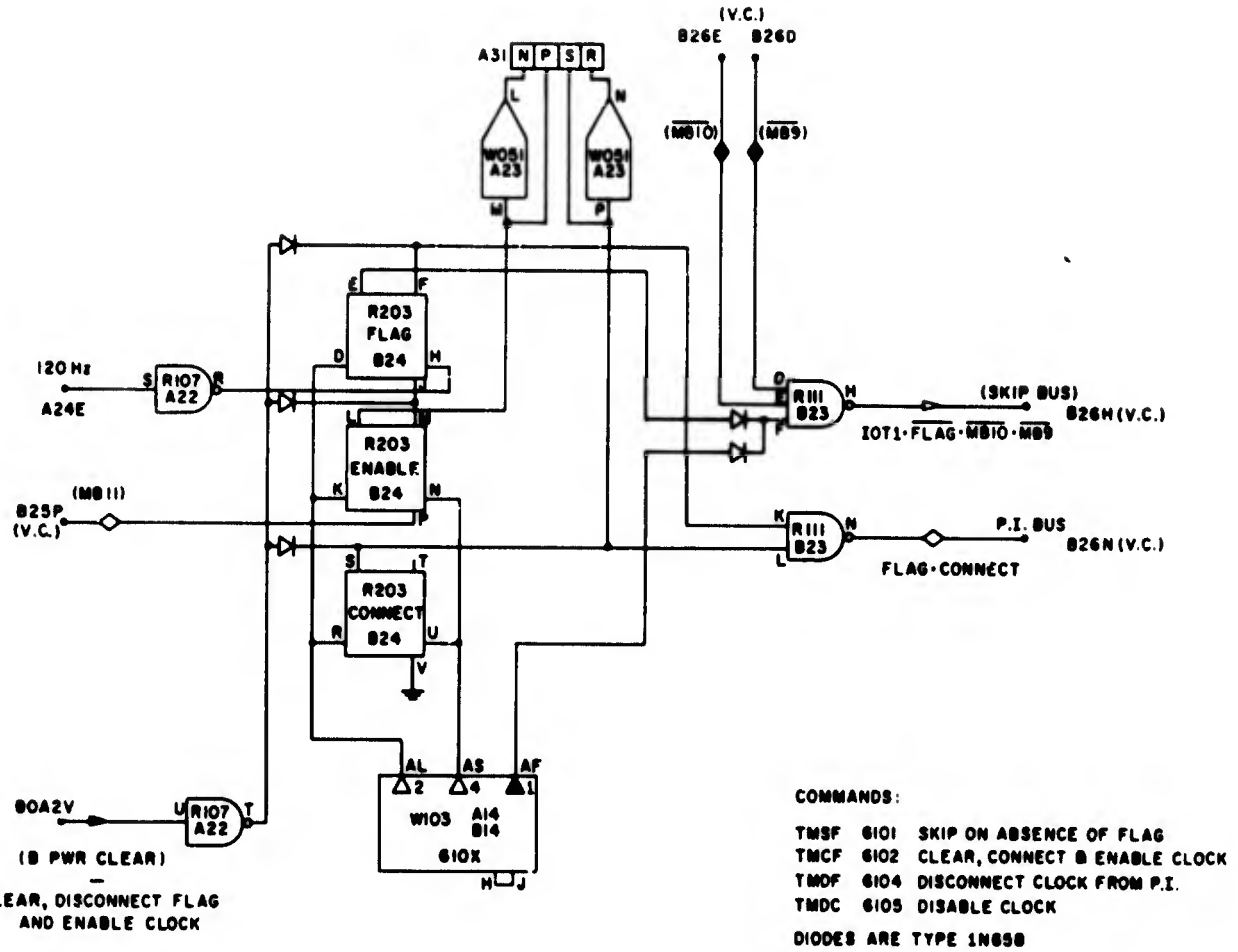


FIGURE 18

120 HZ CLOCK

The output from the flag flip-flop is connected to the two R111 bus gates (at B23). The gate connected to the PI bus (pins K, L and N), is also connected to the output from the connect flip-flop, also a R203 (at B24). When this flip-flop is in the 1 state (as indicated by the connect status light), the connect input to the R111 (pin L) will be at -3 volts, and the set condition of the flag flip-flop will pull the PI bus to 0 volts, requesting a program interrupt. When the connect flip-flop is in the 0 state, the condition of the flag flip-flop will not affect the PI bus.

The other R111 bus gate to which the flag flip-flop is connected is connected to the Skip bus. This gate is used in conjunction with the device selector to test the condition of the flag flip-flop with the TMSF instruction. The gate inputs are connected to the complements of BMB bits 9 and 10 as well as to the flag flip-flop and the device selector module. All inputs to the gate will be at -3 volts (1 state) only when TMSF (6101) is issued by the processor and the flag flip-flop is not in the 1 state. This condition will pull the Skip bus to ground, causing the processor to skip the following instruction. The inputs connected to BMB bits 9 and 10 prevent commands 6103, 6105 and 6107 from being able to cause a skip to occur. This was necessary to prevent a conflict with TMDC, 6105.

The outputs from the enable and connect flip-flops are also connected to light drivers to provide a visual indication of the status of these flip-flops. The indicator lights are push-button lights, and the buttons are connected back to the outputs of the flip-flops to allow them to be cleared manually (an enable switch keeps the flip-flops from

being cleared inadvertently). The lights and switch are on the Clock and Event Channel control box.

The device selector module provides the control pulses needed for program control of the clock. The use of the IOT1 pulse to test the state of the flag has been described previously. The IOT2 pulse resets all flip-flops as follows: flag, clear; enable, set; connect, set. This pulse is not gated in any way. The IOT4 pulse is connected to the pulse inputs to both the enable and connect flip-flops. The connect flip-flop is always reset to the 0 state by this pulse, but the enable flip-flop is reset only if BMB11 is in the 1 state (6105 or 6107). This is all that is needed for decoding the TMDC instruction.

The B PWR CLEAR pulse is also used in this logic to reset all of the flip-flops, but to slightly different states than is done by IOT2 (6102, TMCF). The R107 inverter (pins U and T at A22) inverts the pulse to go to ground, and is connected through isolating diodes to each of the flip-flops. The flip-flops are thus forced to a known state whenever the power is turned on or the START key on the computer is depressed.

5. Variable Clock

Functional Description

The block diagram for the Variable Clock is shown in Figure 19. The clock consists of a variable rate R-C pulse source, flag and control flip-flops, a pulse output to drive the A-D converter (DEC Type AF01A), gating to the Skip and PI busses, light drivers and device selector logic.

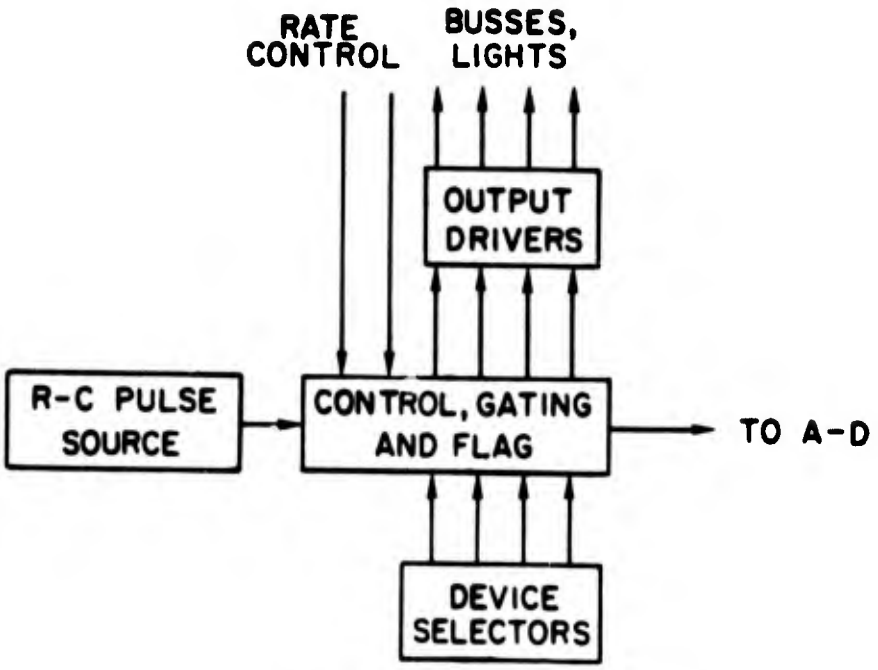


FIGURE 19

VARIABLE CLOCK BLOCK DIAGRAM

The pulse source is a synchronizable multivibrator with a resistor-capacitor timing network. It is enabled and disabled by the enable flip-flop, and when enabled, produces pulses of about 100 nanosecond duration at the rate set by the R-C network. The output from the pulse source goes to the pulse inputs of the flag flip-flop and the A-D converter driver. The select flip-flop and gating determines which of these will respond to the pulses, allowing the clock either to operate as a simple clock for timing purposes or to directly drive the A-D converter for fixed-rate sampling. The flag flip-flop is connected through gating to the Skip and PI busses, allowing program testing of the state of the flag. The connect flip-flop controls the connection of the flag to the PI bus. When in the 1 state, the set condition of the flag will request a program interrupt. The device select logic provides program control of the various clock functions. The light drivers provide an indication of the status of the control flip-flops, and are wired to push-button switch-lights. The switches are wired back to the control flip-flops and allow them to be cleared manually.

Commands

The IOT commands used by the device are from device codes 07 (octal), IOT 6071, and 12 (octal), IOT 612x. The five commands are as follows:

6121 CKSF Skip on Clock Flag not set.

6122 CKCF Clear Clock Flag, enable clock, connect flag to PI bus and select flag.

6124 CKDF Disconnect Clock Flag from PI bus.

6125 CKDC Disable Clock and disconnect flag from PI bus.

6071 CKAD Select A-D driver output.

When it is desired to turn the clock off entirely, CKCF and CKDC are micro-coded together to clear the flag and then disable and disconnect it. The clock is turned back on by giving CKCF, followed by CKAD if it is desired to select the A-D drive output.

Operational Description

The RC pulse source is a stable, synchronizable pulse generator which may be disabled. It is a R401 clock module and has pulse jitter specified as less than 0.2%. When it has been disabled (stopped) for at least one pulse period and is then re-enabled, it will produce a pulse within 70 nanoseconds of the time that the enable line goes to -3 volts. Following pulses will occur at the pulse repetition set by the RC network. This is one of the most important features of the clock, because it allows clock synchronization with external events.

The schematic for the clock is shown in Figure 20. The R401 clock module is enabled by the enable flip-flop section of the R302 (pins K, L, M, N and P at B27). This flip-flop is also connected to the select gating. The connect flip-flop section of the R302 (pins R, S, U and V at B25) is connected to the PI bus output gate, and controls the pulling of the PI bus to ground when the flag flip-flop is set. The select flip-flop section of the R202 (pins D, E, H, J, K and L at A28) controls the use of the pulses from the R401. When this flip-flop is set, outputs from the clock module (which can occur only when it is enabled) will cause

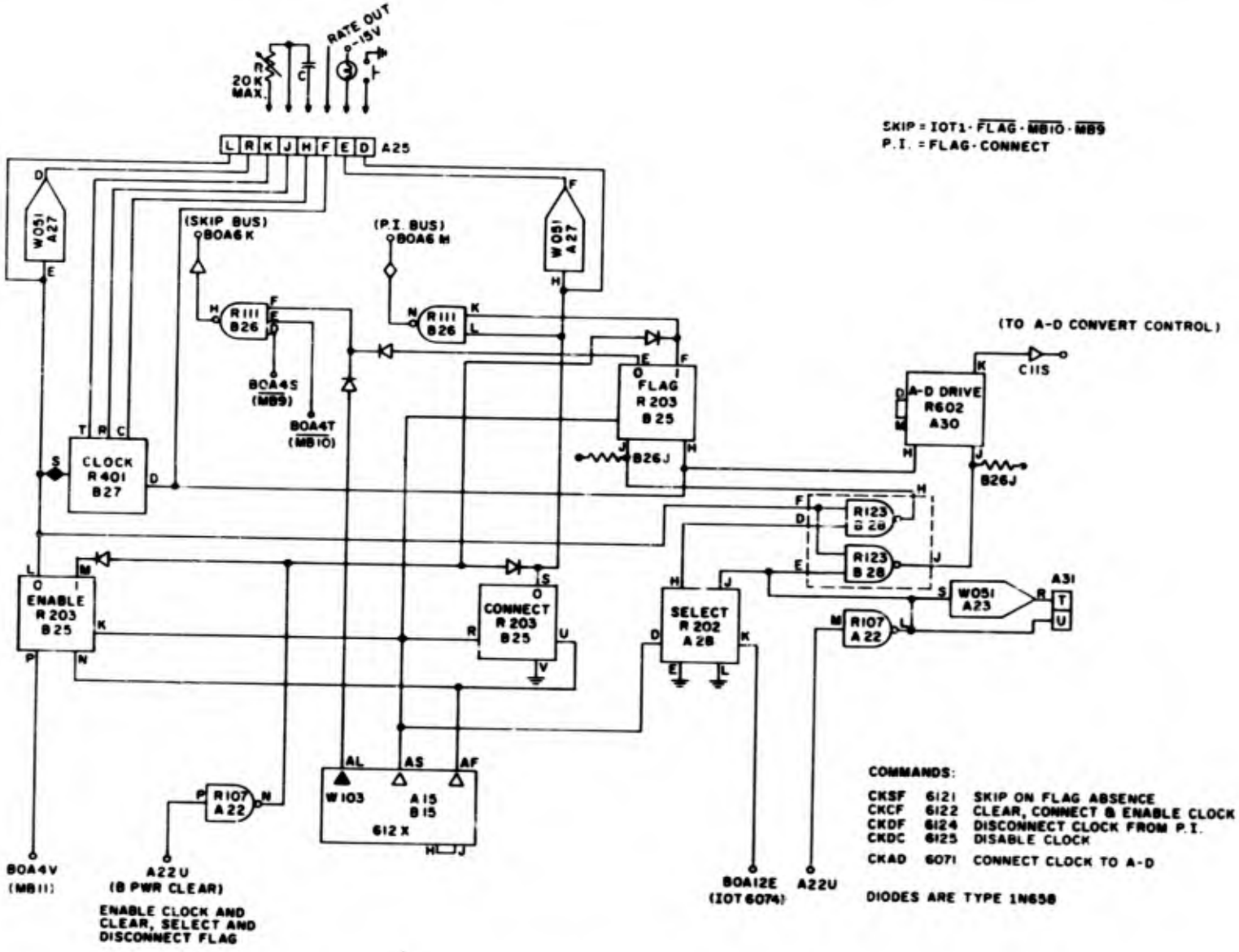


FIGURE 20

VARIABLE CLOCK

the flag flip-flop to be set (which can request a program interrupt if the Connect flip-flop is set). When this flip-flop is clear, the pulses from the clock module will be amplified by the R602 pulse amplifier and cause the A-D converter to begin a conversion.

The flag flip-flop section of the R302 (pins D, E, F, J and H at B25) is used to flag the computer that a clock pulse has occurred (when so selected). It is connected through gating to the Skip and PI busses of the I/O bus, and is reset by the CKCF instruction.

The actual operation of the clock is dependent on the state of the enable flip-flop. This flip-flop, when set to the 0 state (0 volts on pin L at B25), disables the R401 module and no pulses are produced by it. When this flip-flop is set to the 1 state (-3 volts on pin L), the clock module is enabled and will produce pulses at the set rate. These pulses are connected to the rate out terminal for monitoring and to the pulse inputs to the flag section of the R203 and to the R602 pulse amplifier. Depending on the state of the select flip-flop, the pulses will either cause the flag flip-flop to be set or will cause an amplified pulse to be sent to the A-D converter to initiate a conversion. Due to the R123 gates (pins D, E, F, H and J at B28), the enable flip-flop also affects the flag flip-flop and the pulse amplifier. These modules have DCD gates at their inputs which require that their level inputs be at a ground level for at least 400 nanoseconds before pulses at the pulse inputs will be gated through. When the enable flip-flop is in the 0 state, the outputs of both of the R123 gates are at -3 volts, and the DCD gates of the two modules are disabled. When the enable flip-flop

goes to the 1 state, the output of one of the R123 gates (determined by the state of the select flip-flop) will go to 0 volts. At the same time that this is happening, the clock module is being enabled, and produces its first pulse within 70 ns. However, the DCD gates will have been enabled by the ground level for less than 70 ns and will not have been set up completely. Thus the first pulse from the clock module will be ignored. This feature is very important because it allows the desired operation of the clock without any other hardware--the first real pulse from the clock is exactly one clock period from the time that the clock is enabled. This allows, for instance, an A-D conversion to be initiated exactly one clock period from the time that the clock is enabled in response to some event, without any supervison from the computer.

For proper operation of the disable-enable-synchronize feature of the clock, the clock must be disabled for at least one clock period. This is necessary to be sure that the clock module has timed out properly and that the DCD gates have been preset properly.

When the A-D convert pulse is the selected output from the clock, the state of the connect flip-flop will not affect the operation of the clock unless the flag flip-flop is left in the set condition. After it has been cleared, it cannot be set until a CKCF instruction is given to re-select the flag output. Note that since the CKCF instruction not only clears the flag but also re-selects the flag output, any selection of the A-D output by the CKAD instruction must be done after a CKCF is given.

When the flag flip-flop is selected and set, the connect flip-flop controls the pulling of the PI bus to ground through the R111 (pins K, L and N at B26). The action of the flag flip-flop and the control of the flag, enable and connect flip-flops by the W103 device selector (at A15 and B15) is identical to the operation of the 120 Hz clock control, except as described above. Refer to that section for further description of operation.

As with the 120 Hz clock, each of the control flip-flops is connected to a light driver which is used to give a visual indication of the status of the clock. The push-button lights to which they are connected also allow the control flip-flops to be cleared in the same manner as the 120 Hz clock control.

The B PWR CLEAR pulse from the computer is used to reset the control flip-flops to conditions similar to those set up by the CKCF instruction. The R107 inverter (pins P and N at A22) produces a ground-going pulse which is connected to the outputs of the flag, enable and connect flip-flops through isolating diodes. When the pulse is generated by the computer the clock is enabled, the flag is cleared and the flag is disconnected from the PI bus.

6. Event Channels

Functional Description

The block diagram for the Event Channels is shown in Figure 21. Each of the three channels consists of a comparator, a flag flip-flop,

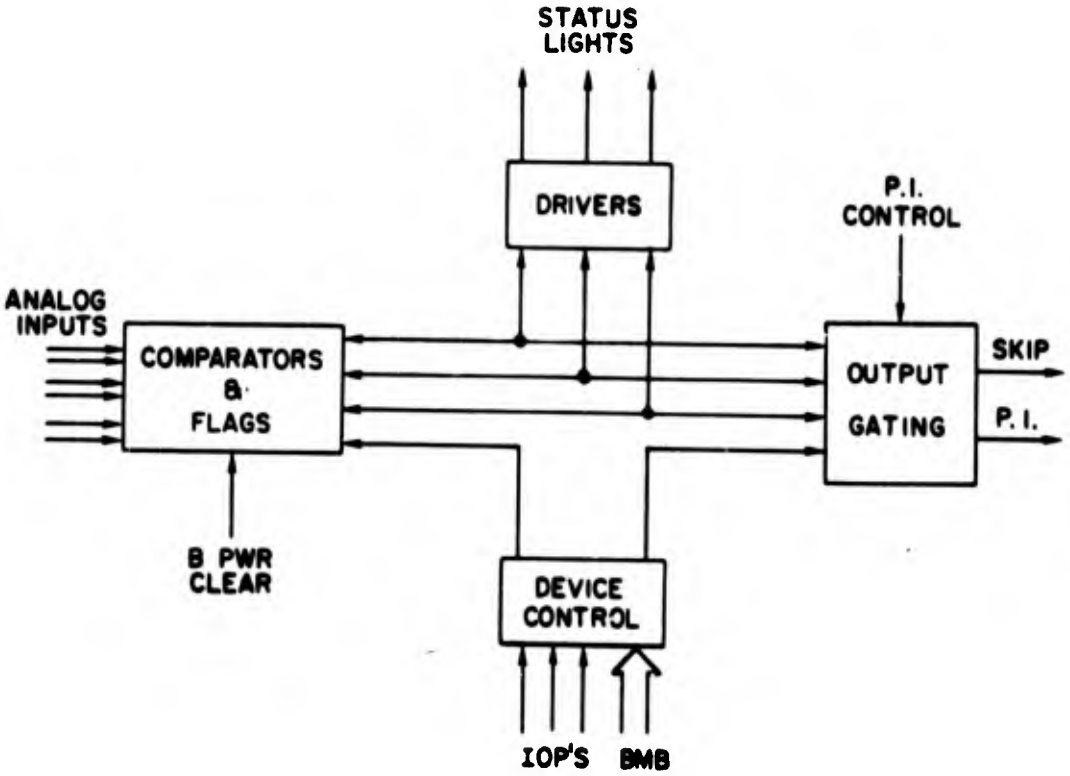


FIGURE 21

EVENT CHANNELS - BLOCK DIAGRAM

Skip and PI bus gating and a flag light driver. All of the flags are connected to or disconnected from the PI bus by a switch on the Clock and Event Channel Control Box, which is mounted in a rack near the computer. When the switch is open, the setting of a flag flip-flop will cause the corresponding gate to pull the PI bus to ground, requesting a program interrupt. When the switch is closed, no program interrupts can be requested.

The lights to which the light drivers are connected are also in the control box. They are push-button lights and the buttons are connected back to the flag flip-flops, allowing the flip-flops to be reset (when so enabled by a switch on the control box).

Commands

The IOT commands used by the device use device codes 13 (octal), IOT 613x, and 14 (octal), IOT 614x. The six commands are as follows:

6131 ECS1 Skip if Event Channel 1 flag is set.

6132 ECS2 Skip if Event Channel 2 flag is set.

6142 ECS3 Skip if Event Channel 3 flag is set.

6141 ECC1 Clear Event Channel 1 flag.

6134 ECC2 Clear Event Channel 2 flag.

6144 ECC3 Clear Event Channel 3 flag.

Operational Description

The schematic for the Event Channels is shown in Figure 22. Since each of the channels is identical except for the controlling IOT pulses, only channel 1 will be described. The comparator is an inexpensive

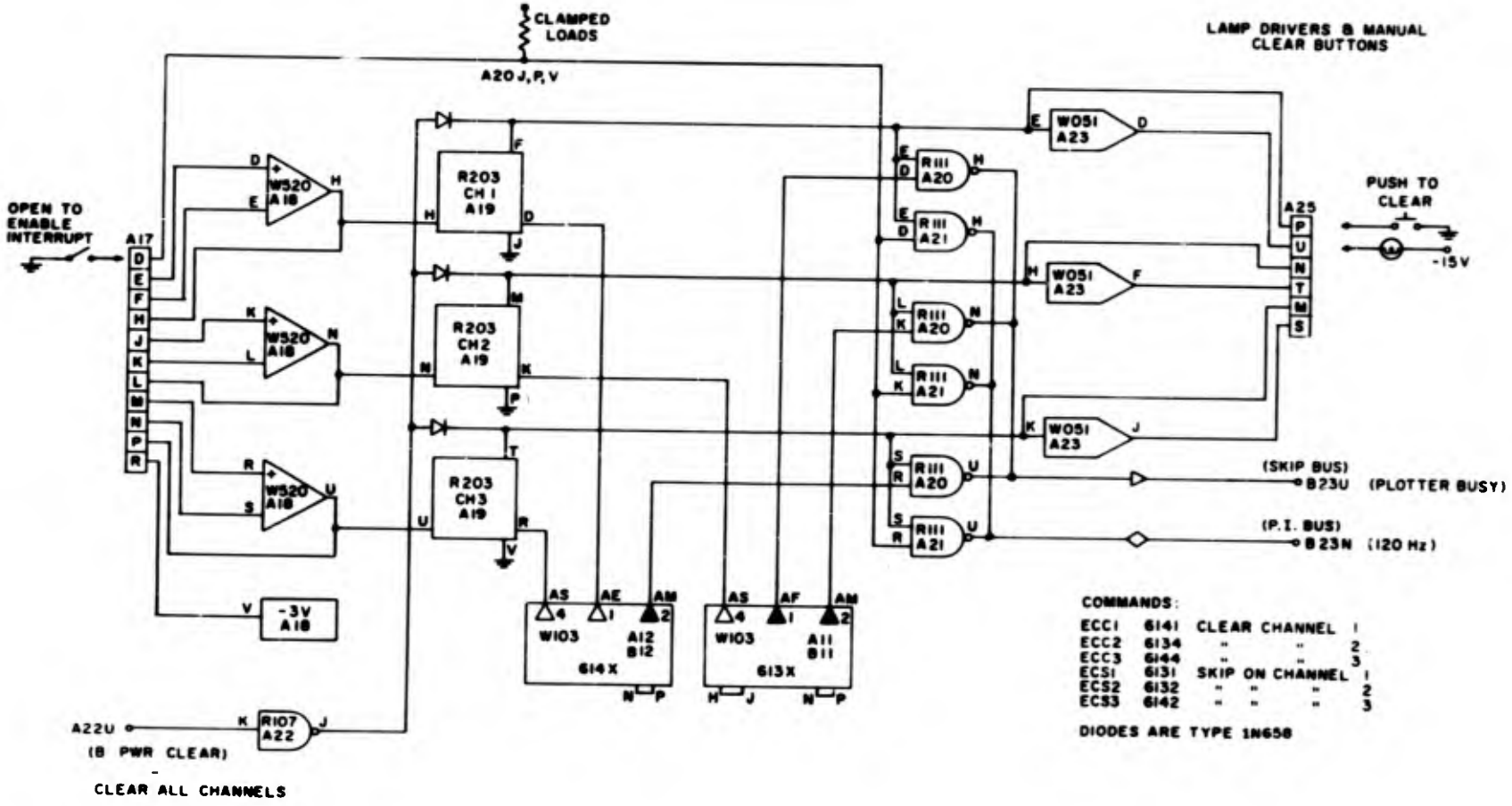


FIGURE 22

EVENT CHANNELS

one which is only tested for operation with input voltage differences of less than 100 millivolts. The output of the W520 comparator (pins D, E and H at A18 for channel 1) will be at 0 volts whenever its "+" input is more positive than the other input. All of the inputs to the comparators come from the Clock and Event Channel Control Box.

The ground-going transition of the comparator output causes the corresponding flip-flop to set. Since the comparator is connected to the pulse input to the R203 flag flip-flop (pin H at A19), the flag can be reset even if the output of the comparator stays at 0 volts. Therefore only a positive-going transition on the input will set the flag.

If the switch on the control box is open, the clamped loads (pins J, P and V at A20) hold one set of the inputs to the gates connected to the PI bus at -3 volts, enabling them. Whenever one of the flags is set, the gate(s) will pull the PI bus to ground, requesting a program interrupt. When the switch is closed, the inputs to the gates are held at 0 volts and are thus disabled.

The program tests the state of a flag by issuing an ECSx command. If the flag flip-flop is set, its output will be at -3 volts, enabling the corresponding R111 gate (at A20). When the IOT pulse is sent out by one of the device selectors, it will be gated through to the Skip bus which it will pull to ground, causing the processor to skip the next sequential instruction.

The flag can be reset by issuing an ECCx command. The IOT pulse from the device decoder pulls the reset input of the flip-flop

to 0 volts, directly resetting it. As mentioned above, the flag can also be reset by pushing the corresponding push-button light on the control box. Doing this forces the flip-flop into the reset state by forcing the output from the flip-flop to 0 volts.

The B PWR CLEAR pulse from the computer forces all flag flip-flops to the reset state. The R107 inverter (pins K and J at A22) produces a ground-going pulse which is connected to the outputs of the flip-flops through isolating diodes. This pulse forces the outputs of the flip-flops to 0 volts, resetting them.

7. Plotter Control and Utility Pulses

Functional Description

The block diagram for the Plotter Control and Utility Pulses is shown in Figure 23. The plotter control consists of two pulse generators which are used by the X-Y plotter paper advance, gating to test the status of the plotter paper advance, and a flip-flop and driver to control the plotter pen (raise and lower it). The Utility pulses are generated by two other pulse generators and are available for use as desired.

The X-Y recorder is driven by the Digital-to-Analog (D-A) converters in the oscilloscope control (DEC Type 34D). Since two oscilloscopes and the X-Y recorder are connected to these outputs with a considerable amount of cable, it was also necessary to add booster operational amplifiers between the outputs of the D-As and the cables. They are also shown in the block diagram.

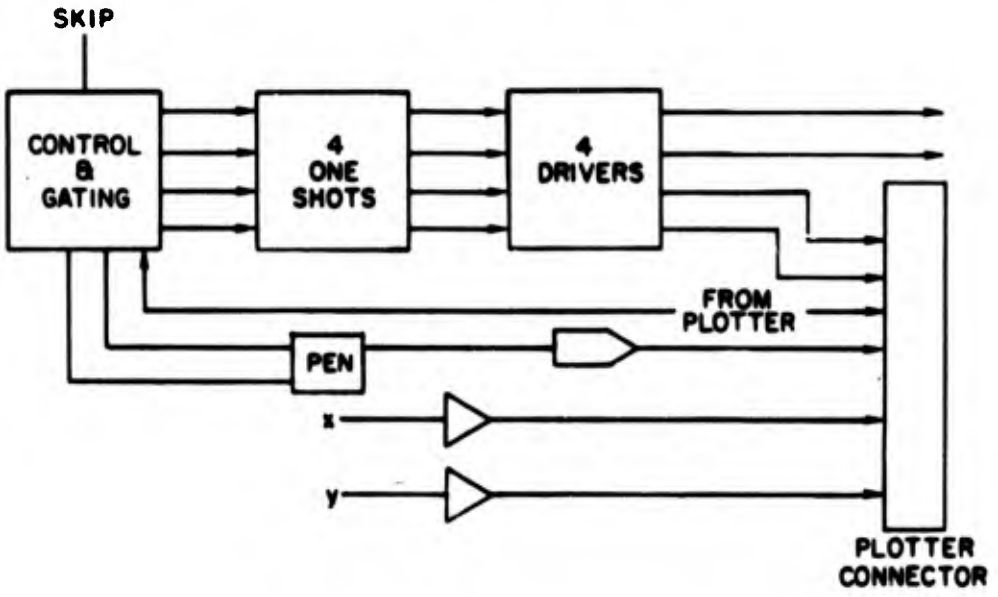


FIGURE 23

PLOTTER CONTROL & UTILITY PULSES

BLOCK DIAGRAM

Commands

The IOT commands for these devices use device codes 02 (octal), IOT 602x, 07 (octal), IOT 6072, and 30 (octal), IOT 630x. The seven commands are as follows:

6021 PMDA Plotter Major Division Advance.

6022 PAIM Plotter Incremental Advance.

6024 SDP Set Display (plotter) pen (lower it).

6072 CDP Clear Display (plotter) pen (raise it).

6301 PSOB Plotter Skip on Busy (chart advance).

6302 UPP1 Utility Pulse 1.

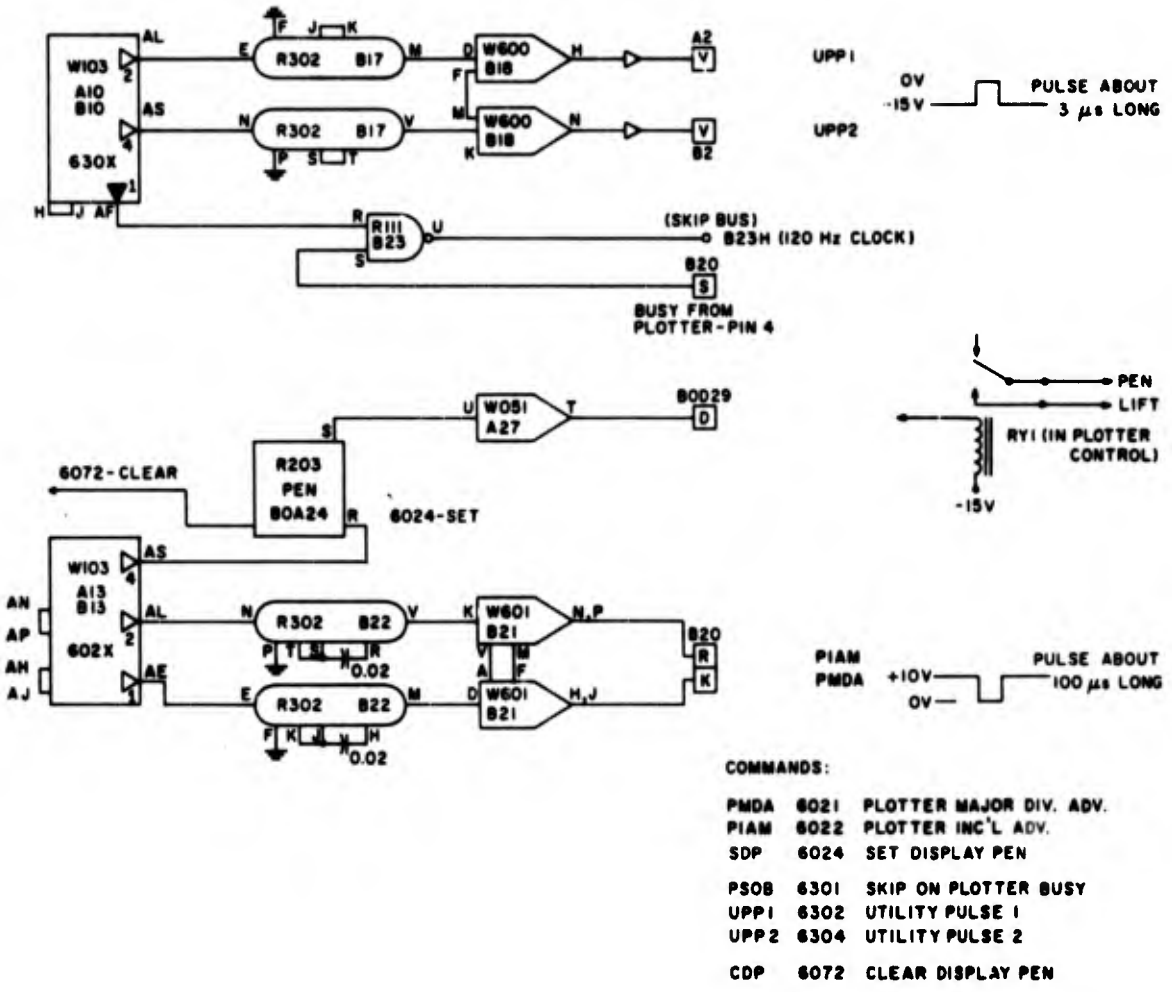
6304 UPP2 Utility Pulse 2.

PSOB is used to test for paper in motion after either PMDA or PIAM is issued.

Operational Description

The schematic for the Plotter Control and Utility Pulses is shown in Figure 24. The plotter control consists of the W103 device selector (at A13 and B13), the R302 dual one-shot (at B22), the W601 positive output converter (at B21), the R203 pen flip-flop (pins R and S at BOA24), the WO51 relay driver (pins U and T at A27) and the R111 gate (pins R, S and U at B23) which is used to drive the Skip bus.

The plotter control pulses are derived by simply stretching the IOT pulses output by the device selector (at A13 and B13) to about



COMMANDS:

PMDA	6021	PLOTTER MAJOR DIV. ADV.
PIAM	6022	PLOTTER INC'L ADV.
SDP	6024	SET DISPLAY PEN
PSOB	6301	SKIP ON PLOTTER BUSY
UPP1	6302	UTILITY PULSE 1
UPP2	6304	UTILITY PULSE 2
CDP	6072	CLEAR DISPLAY PEN

FIGURE 24

PLOTTER CONTROL & UTILITY PULSES

100 microseconds with the one shots and then converting them to the positive levels needed by the plotter chart advance. The pulse caused by the PIAM command is connected to the incremental advance input in the chart advance control and causes the paper to be moved one increment forward (as set by a switch on the advance control). The pulse caused by the PMDA command is connected to the major division advance input and moves the paper one inch.

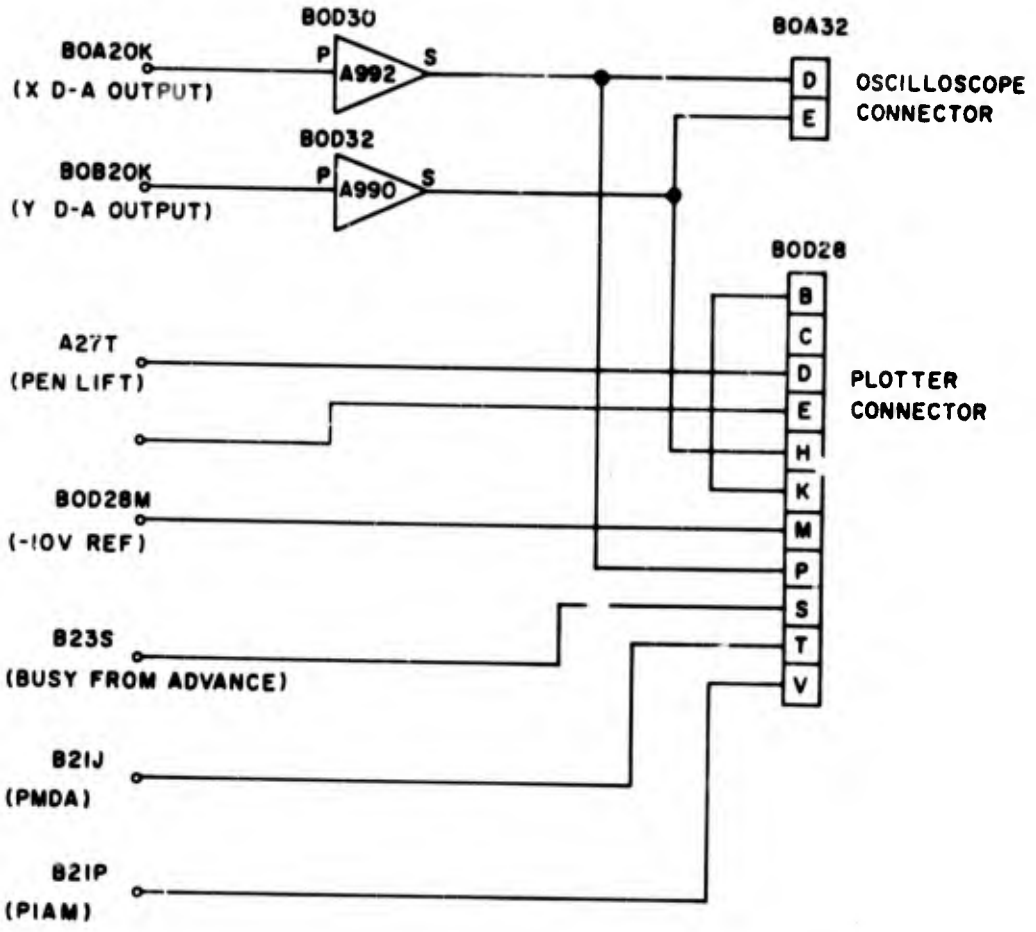
The status of the plotter paper advance is tested by the PSOB command. The 6301 IOT pulse from the device selector (pin A10F) is simply gated by the R111 gate (pins R, S and U at B23) to the Skip bus when the paper advance busy line is negative. When the advance is not busy, the line is at 0 volts and the IOT will not be gated through.

The plotter pen control uses a R203 flip-flop (pins S and R at BOA24) which is part of the 34D display control, and which was intended to be used for a light pen flag. Thus part of the circuit is not shown in Figure 3-17, but will be found in the 34D prints. The output from the flip-flop was simply re-routed to a relay driver which is used to drive an external relay which, in turn, raises and lowers the pen on the X-Y plotter. The flip-flop is set by the SDP command, which causes an IOT pulse from the device selector (pin A13S) to set the flip-flop through the direct set input. The flip-flop is cleared by either the CDP command or by the B PWR CLEAR pulse through gating in the 34D control.

The utility pulses are generated the same that the plotter control pulses are generated, with the exception that a W600 negative output

converter (at B18) is used rather than the W601 and the pulses are stretched to only 3 microseconds rather than 100. These two pulse outputs were designed to drive a special 100 KHz multiplexer designed and built by the Tropospheric Propagation Group which is described in other reports.[5,6]

As mentioned above, X and Y booster amplifiers were added to the outputs of the X and Y D-A converters in the 34D display control. The capacitive load of the cables connecting the outputs to the two display oscilloscopes and the X-Y recorder caused severe delays in the settling time of the D-A converters. The result was that the voltage on the cables would still be settling at the time that the intensify pulse from the 34D reached the display oscilloscopes, producing a smeared display. The connection of the booster amplifiers is shown in Figure 25 along with the connections to the X-Y plotter cable connector.



X & Y BOOSTER AMPLIFIERS: UNITY GAIN, NON-INVERTING

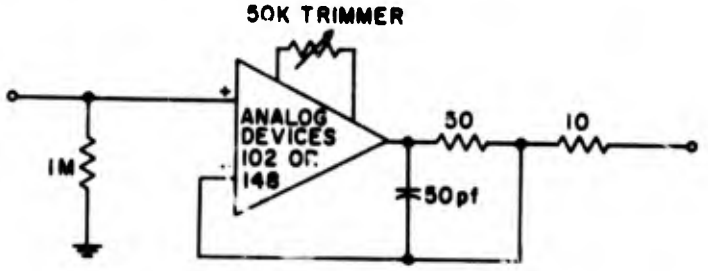


FIGURE 25

X & Y BOOSTER AMPLIFIERS & PLOTTER CONNECTOR

IV. CONCLUSION

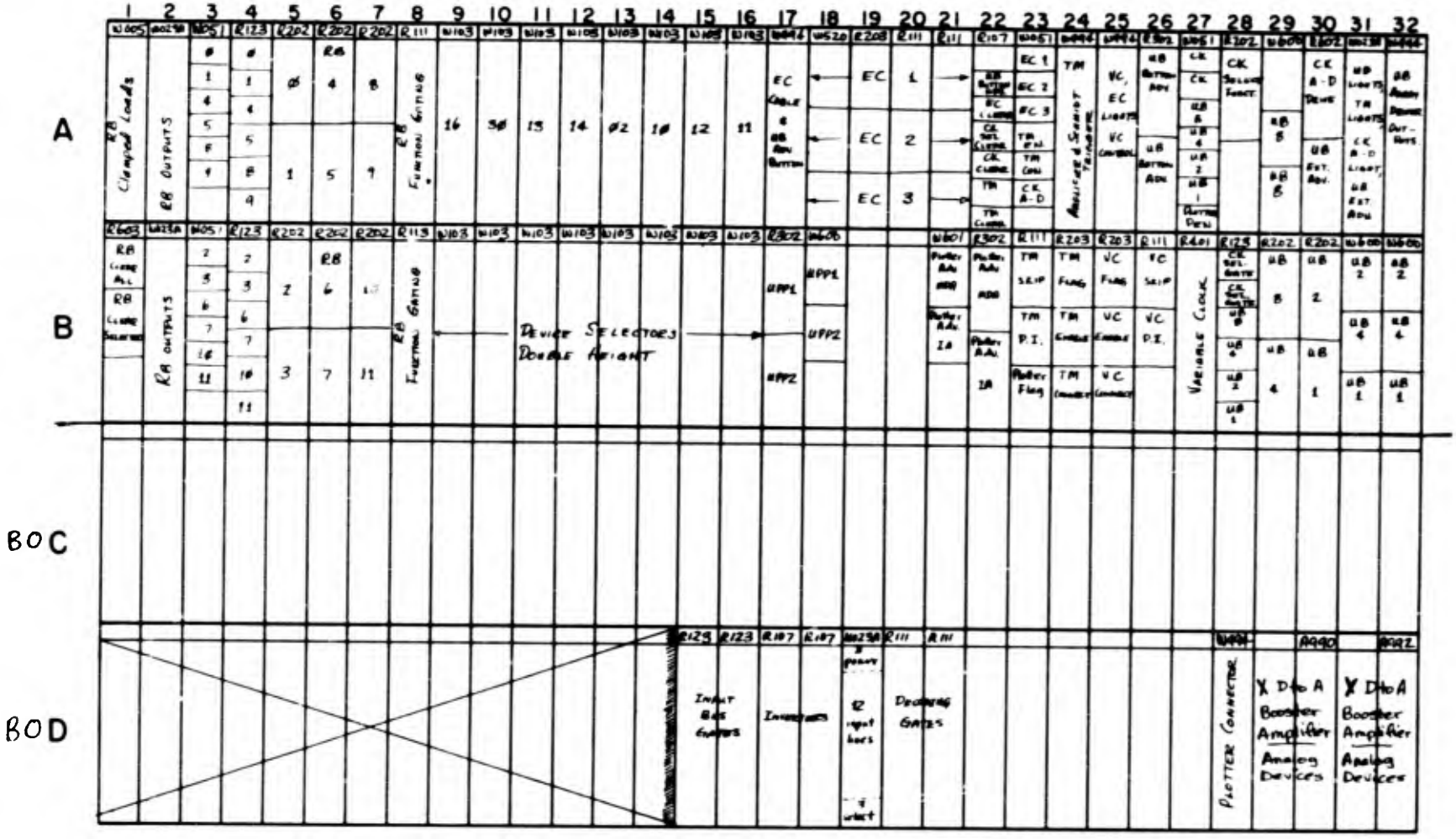
The design checkout for the devices was accomplished in three stages. The devices were first built and bench tested away from the computer. As mentioned in the introduction, this was primarily to reduce the time that the computer would be non-operational during device installation. The devices were then installed and tested, with some debugging, until they were in operational condition. Finally the data collection program that had prompted the hardware design was finished and the hardware and software were fine tuned together. It was during this time that the need for the extra switch registers was discovered, so they were designed, installed and tested also. The data acquisition program was then updated to use the extra switches.

The complete system, including the devices described previously, the IBM compatible magnetic tape unit and the data acquisition program, have been tested carefully and have met the original design goals. Furthermore, the system has been tested at an aggregate data rate of 7,200 words per second onto the magnetic tape, which is about 20 percent faster than the maximum data rate which was predicted to be needed. The data acquisition program is more thoroughly described in an unpublished manuscript.[4]

Since the time that the system has become operational, all of the data acquisition using the three Tropospheric Propagation Group arrays has been done using this system. In the first year of operation of the system, more than twenty 2400 foot reels of data have been acquired.

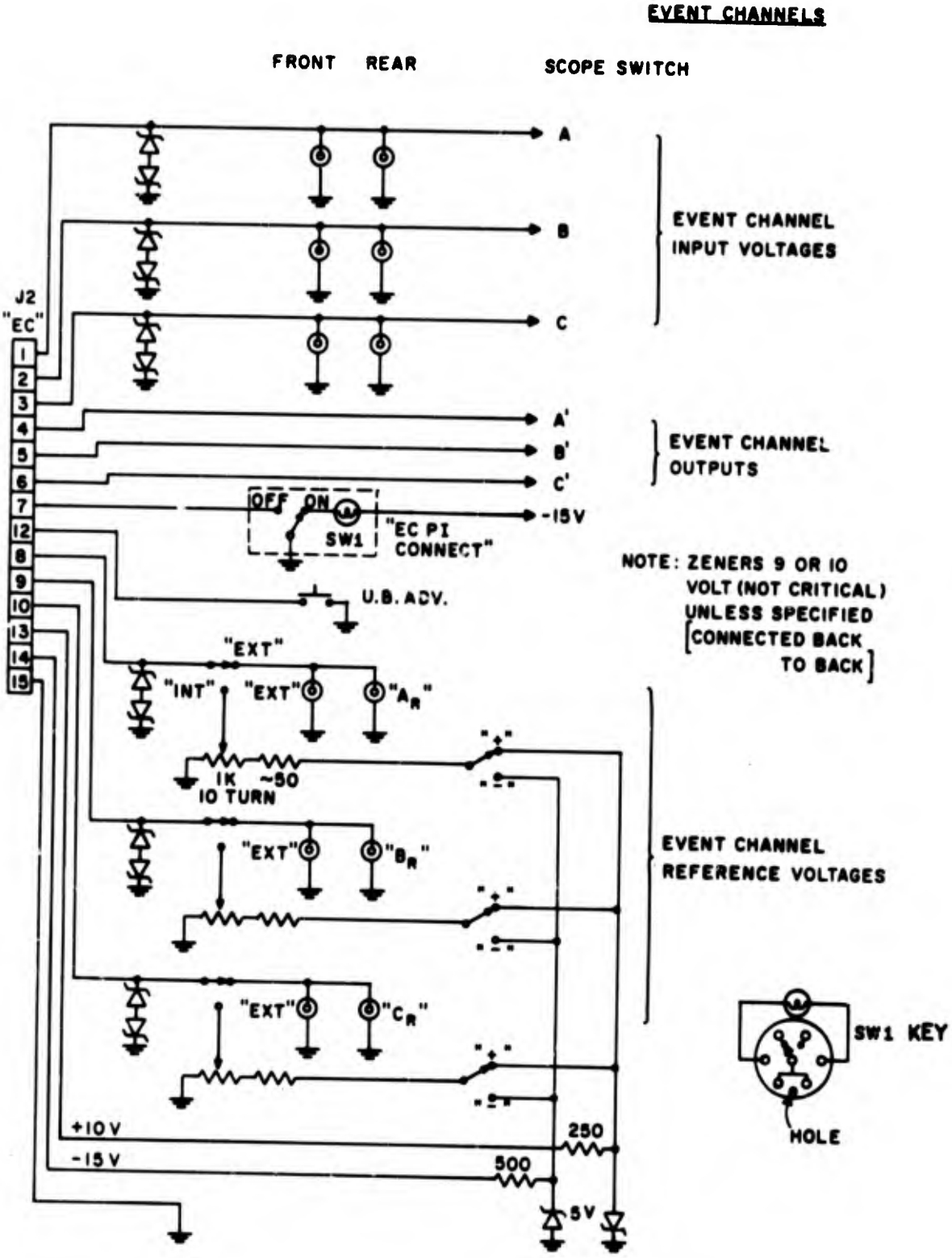
This is equivalent to over 200 million data samples, more data than had been acquired in the previous 4 years. The data so acquired and its analysis has been discussed in the Ph.D. theses of Drs. Peter Mandics,[5] Jeff Harp[6] and Nick Cianos[7] and in a final report by Dr. Alan Waterman.[8]

Since the time that the system has been operational, most of the other programs used by the Group (mostly data reduction programs) have also been modified to make use of the new hardware and the IBM compatible tape unit, with a tremendous improvement in system throughput.



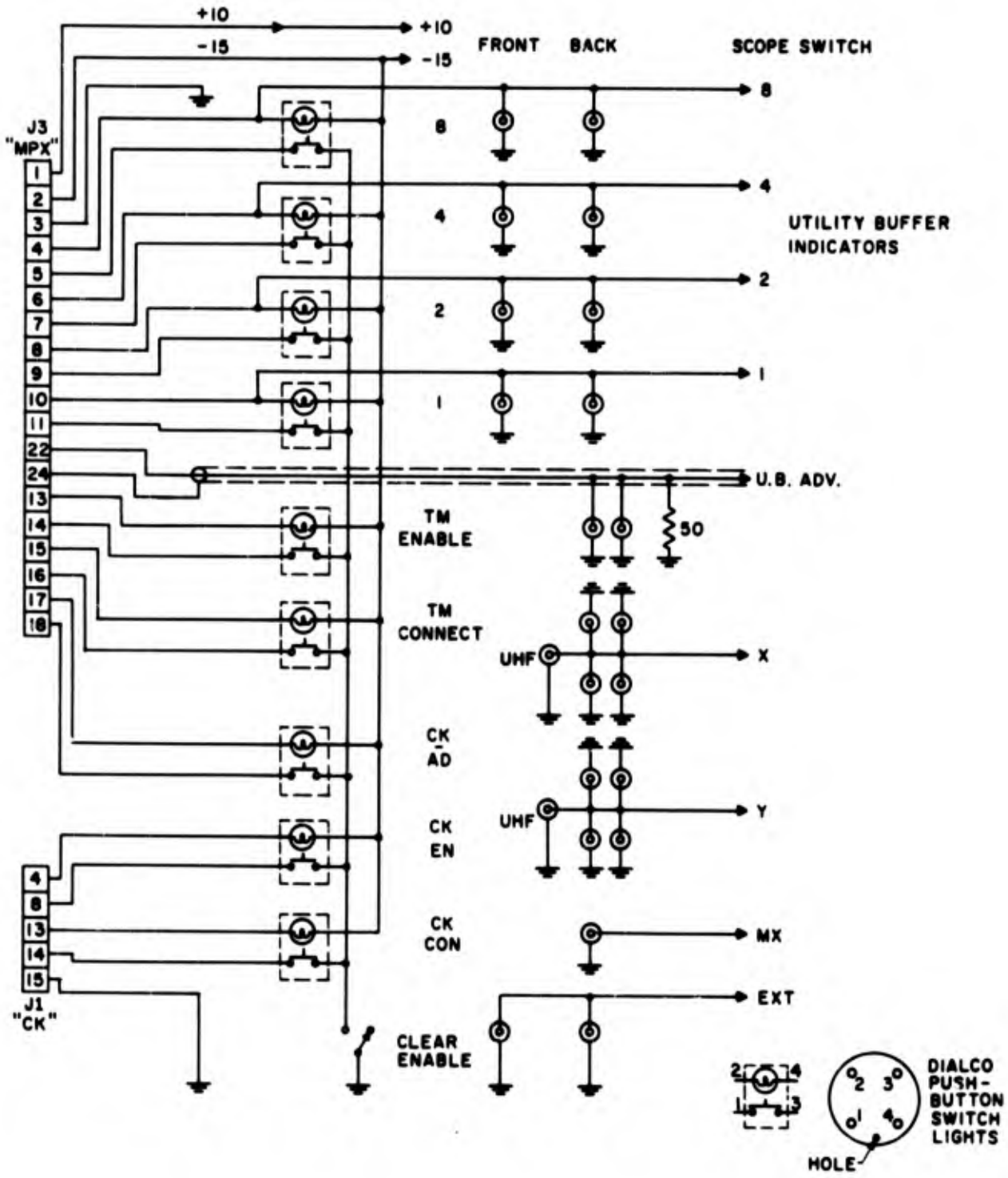
APPENDIX A.

MODULE UTILIZATION DRAWING



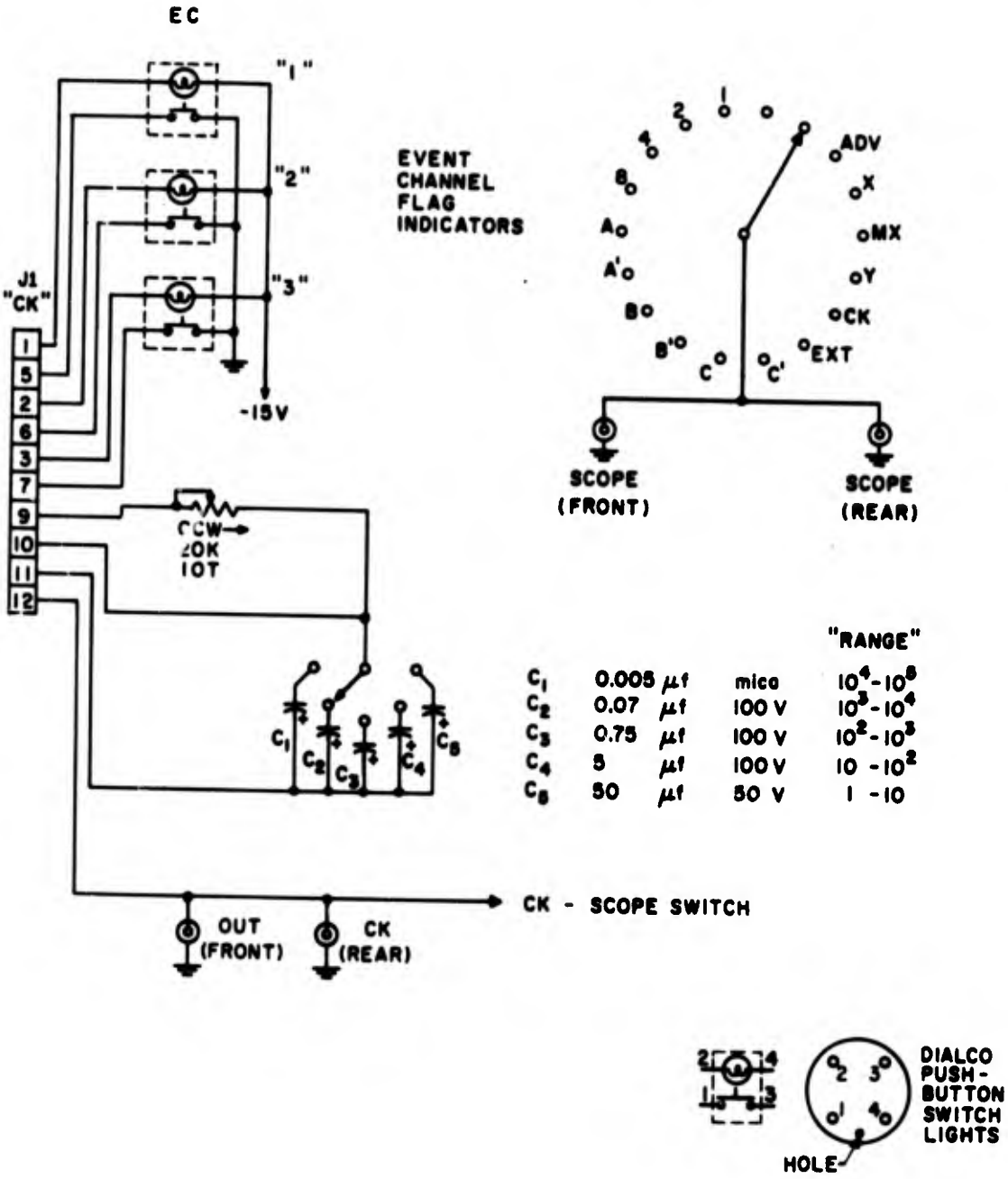
APPENDIX B.

CLOCK & EVENT CHANNEL CONTROL BOX--PART 1



APPENDIX C.

CLOCK & EVENT CHANNEL CONTROL BOX--PART 2



APPENDIX D.

CLOCK & EVENT CHANNEL CONTROL BOX--PART 3

BIBLIOGRAPHY

1. Small Computer Handbook, Digital Equipment Corp., Maynard, Mass. 1967.
2. Logic Handbook, Digital Equipment Corp., Maynard, Mass. 1967.
3. Introduction to Programming, Digital Equipment Corp., Maynard, Mass., 1970.
4. Unpublished manuscript, the writeup for Rapid, Versatile Data Gathering Program, by Jim Crapuchettes.
5. Mandics, P.A., "Line-of-Sight Acoustical Probing of Atmospheric Turbulence," Stanford Electronics Laboratories, Stanford University, SU-SEL-71-002 (March 1971).
6. Harp, J.C., "A Line-of-Sight Microwave Propagation Experiment for Resolving the Motions and Turbulent Structure of the Atmosphere," Stanford Electronics Laboratories, Stanford University, SEL-71-042 (August 1971).
7. Cianos, N., "Angle and Doppler Measurements of Microwaves Propagated Beyond the Horizon," Stanford Electronics Laboratories, Stanford University, SU-SEL-70-069 (October 1970).
8. Waterman, A.T., Jr., "Transhorizon Propagation Techniques," Final Report, Stanford Electronics Laboratories, Stanford University, SU-SEL 71-010 (April 1971).